

Université des Sciences et de la Technologie d'Oran
Faculté des Mathématiques et Informatique
Département d'informatique

Polycopié de cours

Intitulé

Structure des machines 2

Auteur

Dounia YEDJOUR

Ce cours est destiné aux étudiants de première année licence math-info

Table des matières

Table des matières.....	2
Liste des figures	5
Liste des tableaux.....	6
Chapitre 1 :.....	7
Algèbre de Boole	7
1 Introduction.....	7
2 Postulas de l’algèbre de Boole.....	7
3 Réalisation d’un schéma logique.	8
3.1 Portes logiques.....	8
3.2 Les opérateurs NAND et NOR.....	9
3.2.1 Opérateur NAND (Not AND)	9
3.2.2 Opérateur NOR (Not OR)	11
4 Table de vérité (TV).....	12
4.1 Extraction des fonctions logiques à partir de la TV (FORMES CANONIQUES).....	12
4.1.1 Première forme canonique (forme disjonctive).....	12
4.1.2 Deuxième forme canonique (conjonctive):	13
5 Simplification des fonctions logiques.....	14
5.1 Simplification Algébrique :	14
5.2 Simplification par karnaugh :	15
5.2.1 Création de la table de karnaugh (TK) :.....	15
5.2.2 Remplissage de la table de karnaugh :	16
5.2.3 Notion des cases adjacentes :	18
5.2.4 Simplification par karnaugh :	19
5.2.5 Simplification par karnaugh à partir d’une table incomplète:	21
6 Conclusion :	21
7 Exercices	22

Chapitre 2.....	27
Les circuits combinatoires	27
1 Introduction.....	27
2 Etapes de conception d'un circuit combinatoire:.....	27
3 Etude de quelques circuits combinatoires usuels :.....	28
3.1 Le demi-additionneur (DA):.....	28
3.2 L'additionneur complet :	29
3.3 Le Demi-soustracteur (DS) :.....	32
3.4 Le soustracteur complet :.....	33
3.5 Le comparateur :.....	34
3.6 Le codeur (ou encodeur) :.....	35
3.7 Le décodeur :	38
3.8 Le Transcodeur :.....	39
3.9 Le multiplexeur :.....	41
3.10 Le démultiplexeur	43
4 Conclusion	44
5 Exercices	45
Chapitre 3.....	48
La logique séquentielle.	48
1 Introduction.....	48
2 Types de circuits séquentiels :	49
2.1 Circuits séquentiels asynchrone.....	49
2.1.1 La bascule RS :.....	49
2.1.2 La bascule JK :	52
2.1.3 La bascule D :.....	54
2.2 Circuits séquentiels synchrones.....	55
2.2.1 Horloge :.....	55
2.2.2 Type de synchronisation :.....	56
2.2.3 La bascule RS synchrone (RSH).....	58
2.2.4 La bascule JK synchrone (JKH).....	59
2.2.5 La bascule D synchrone:	61
3 Les registres	63
3.1 Définition.....	63
3.2 Fonctionnement des registres	63
3.2.1 Registre à décalage (Registre série):	63
3.2.2 Registre à chargement parallèle :	65

4	Analyse d'un circuit séquentiel (automates):.....	66
4.1	Définition :.....	66
4.2	Table de transition	66
4.3	Automate de Mealy.....	67
4.4	Automate de Moore	70
5	Les compteurs / décompteurs:.....	74
6	Conclusion :	78
7	Exercices	79
	Corrigé des Exercices.....	82
1	Chapitre 1	82
2	Chapitre 2.....	91
3	Chapitre 3.....	105
	Références :.....	115

Liste des figures

Figure 1: Logigramme d'un demi-Additionneur.	29
Figure 2 : Boîte noire d'un additionneur complet.	29
Figure 3: schéma d'addition de deux nombres de 4 bits.	30
Figure 4: Logigramme d'un additionneur complet.	31
Figure 5: Additionneur complet réalisé à partir des demi-additionneur.	32
Figure 6: Logigramme d'un demi-soustracteur.	33
Figure 7: Logigramme d'un soustracteur complet 1 bit.	34
Figure 8: Boîte noire d'un comparateur 1 bit.	34
Figure 9: Circuit logique d'un comparateur 1 bit.	35
Figure 10: Schéma du codeur decimal-BCD.	36
Figure 11: Circuit logique du codeur decimal-BCD.	37
Figure 12: Circuit logique du décodeur 3 vers 8.	39
Figure 13: Circuit logique d'un transcodeur BCD – XS-3.	41
Figure 14: Schéma d'un multiplexeur.	42
Figure 15: Circuit logique d'un multiplexeur 4 vers 1.	43
Figure 16: Circuit logique d'un démultiplexeur 1 vers 4.	44
Figure 17: Shéma d'un circuit séquentiel.	48
Figure 18: Boîte noire de la bascule RS asynchrone.	49
Figure 19: Logigramme de la bascule RS.	51
Figure 20: Logigramme de la bascule JK.	54
Figure 21: Réalisation de la bascule D en fonction de la bascule JK.	55
Figure 22: Signal périodique d'une horloge.	55
Figure 23: Logigramme de la bascule RS en front montant.	58
Figure 24: Logigramme de la bascule RS en mode descendant.	59
Figure 25: Logigramme de la bascule JK synchrone.	59
Figure 26: Logigramme de la bascule D synchrone.	62
Figure 27: Registre à décalage à droite à 4 bits.	64
Figure 28: Registre à décalage à gauche à 4 bits (04 bascules).	64
Figure 29: Registre à chargement parallèle à 4 bits.	65
Figure 30: schéma d'un compteur synchrone modulo 4 par les bascules D.	76
Figure 31: schéma d'un décompteur synchrone modulo 8 par les bascules D.	78

Liste des tableaux

Tableau 1 : Postulas de l'algèbre de Boole.....	8
Tableau 2 : Les portes logiques.	9
Tableau 3 : Extraction des formes canoniques à partir de la table de vérité.....	13
Tableau 4 : La table de vérité du décodeur 3 vers 8	38
Tableau 5 : La table de vérité du démultiplexeur 1 vers 4.....	43
Tableau 6 : Fonctionnement de la bascule RS	50
Tableau 7 : Fonction simplifiée sous forme SOP	51
Tableau 8 : Fonction simplifiée sous forme POS	51
Tableau 9 : Fonctionnement de la bascule JK	52

Chapitre 1 :

Algèbre de Boole

1 Introduction

L'algèbre de boole mise en oeuvre par le mathématicien anglais George BOOLE (1815-1864) est une algèbre basée sur les deux valeurs logiques (0 et 1), **0** signifie le « Faux ou False » et **1** signifie le « vrai ou true). Trois (03) opérateurs de bases sont utilisés dans l'algèbre de boole :

- Opérateur ET (en anglais AND), représenté aussi par le symbole « . »,
- Opérateur OU (en anglais OR), représenté aussi par le symbole « + »
- Opérateur NON (en anglais NOT), représenté aussi par le symbole « _ »

Les propriétés relatives aux 03 opérateurs de base (ET, OU et Non) sont résumées dans le tableau 1.

2 Postulas de l'algèbre de Boole

soit ρ un ensemble de variables à deux états, de valeurs de vérité 1 (vrai), 0 (faux), muni d'un nombre limité d'opérateurs : NON($\bar{\quad}$), ET(X) , OU (+). ρ est une algèbre de Boole si les postulats suivants sont vérifiés [1-2] : Soit a, b et c trois variables booléennes

<p>Loi de Commutativité $a + b = b + a$ $a \cdot b = b \cdot a$</p> <p>Associativité $a.(b.c) = (a.b).c = a.b.c$ $a+(b+c) = (a+b)+c = a+b+c$</p> <p>Distributivité $a+ (b \cdot c) = (a + b) \cdot (a + c)$ $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$</p>	<p>Identité (élément neutre) $a + 0 = a$ $a \cdot 1 = a$</p> <p>Complémentarité $a + \bar{a} = 1$ $a \cdot \bar{a} = 0$ $\bar{\bar{a}}$ est le complément de a</p> <p>Idempotence $a + a = a$ $a \cdot a = a$</p>	<p>Absorption $a + 1 = 1$ $a \cdot 0 = 0$</p> <p>Lois de Morgan $\overline{a + b} = \bar{a} \cdot \bar{b}$ $\overline{a \cdot b} = \bar{a} + \bar{b}$</p> <p>Loi d'involution $\overline{\bar{a}} = a$</p>
--	--	---

Tableau 1 : Postulas de l'algèbre de Boole

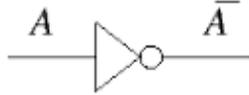
3 Réalisation d'un schéma logique.

Une expression algébrique ou une fonction logique est définie par un ensemble de variables logiques reliées entre elles par des opérateurs logiques. Exemple $F(x,y,z) = x.y + \bar{x}z$.

Un schéma logique ou un circuit logique ou combinatoire est un ensemble de portes logiques reliées entre elles pour répondre à une expression algébrique [3-4].

Une porte logique est un circuit électronique élémentaire qui permet de réaliser la fonction d'un opérateur logique de base. Le tableau 2 résume les portes logiques existantes. Connaissant les portes logiques, on peut construire immédiatement le logigramme de la fonction. Les variables d'entrées correspondent aux fils (lignes) électriques [4-5].

3.1 Portes logiques

Opérateurs	Fonctions élémentaires	Portes
NOT (Inverseur)	\bar{x}	
AND	$A.B$ $(A \wedge B)$	

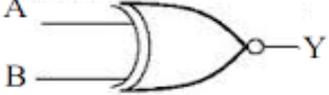
OR	$A+B$ $(A \vee B)$	
NAND	$\overline{x \cdot y}$ $\overline{(A \wedge B)}$	
NOR	$\overline{x + y}$ $\overline{(A \vee B)}$	
XOR	$A \oplus B$	

Tableau 2: Les portes logiques.

3.2 Les opérateurs NAND et NOR

3.2.1 Opérateur NAND (Not AND)

La fonction suivante est le résultat de l'application de l'opérateur NAND sur les deux variables logiques a et b :

$$F(a, b) = \overline{a \cdot b}$$

L'opérateur NAND est un opérateur complet car il permet la réalisation des trois fonctions de base (ET, OU et NON) [4].

Démonstration :

1) Exprimer la fonction NON par des opérateur NAND

Soit la fonction f telle que $f = \overline{a}$

En appliquant la loi d'idempotence ($x = x \cdot x$), nous aurons $\overline{a} = \overline{a \cdot a}$

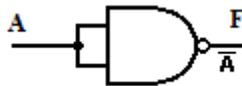
En remplaçant dans f, nous obtenons :

$$f = \overline{a} = \overline{a \cdot a}$$

nous avons pu exprimer la fonction $f = \overline{a}$ à l'aide d'un opérateur NAND.

$$f = \overline{a} = \text{NAND}(a, a)$$

le circuit correspondant est le suivant :



2) Exprimer la fonction ET par des opérateur NAND

Soit la fonction f telle que $f = a \cdot b$

En utilisant la loi d'involusion sur f, nous aurons :

$$f = \overline{\overline{f}} = \overline{\overline{a \cdot b}}$$

En appliquons la loi d'idempotence ($x = x \cdot x$), nous aurons $\overline{a \cdot b} = \overline{a \cdot b \cdot a \cdot b}$

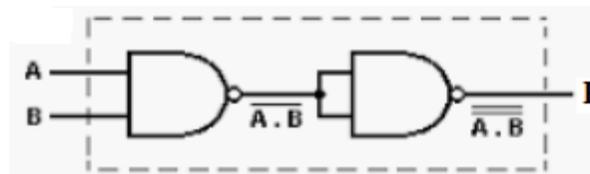
En remplaçons dans f, nous obtenons :

$$f = \overline{\overline{f}} = \overline{\overline{a \cdot b}} = \overline{\overline{a \cdot b \cdot a \cdot b}}$$

nous avons pu exprimer la fonction $f = a \cdot b$ à l'aide de trois opérateur NAND.

$$f = a \cdot b = \text{NAND}(\text{NAND}(a, b), \text{NAND}(a, b))$$

le circuit correspondant est le suivant :



3) Exprimer la fonction OU par des opérateur NAND

Soit la fonction f telle que $f=a+b$

En utilisant la loi d'involusion sur f, nous aurons :

$$f = \overline{\overline{f}} = \overline{\overline{a + b}}$$

en appliquant la loi de Morgan, nous obtenons :

$$f = \overline{\overline{a + b}} = \overline{\overline{a} \cdot \overline{b}}$$

En appliquant la loi d'idempotence ($x=x.x$), nous aurons $\overline{a} = \overline{a \cdot a}$ et $\overline{b} = \overline{b \cdot b}$

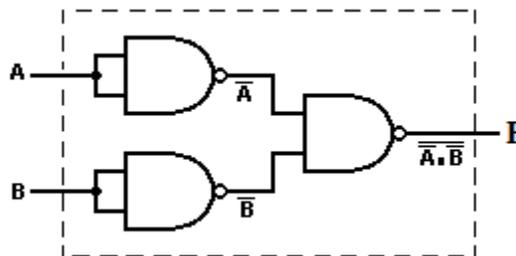
En remplaçant dans f, nous obtenons :

$$f = \overline{\overline{a + b}} = \overline{\overline{a} \cdot \overline{b}} = \overline{\overline{a \cdot a} \cdot \overline{b \cdot b}}$$

nous avons pu exprimer la fonction $f = a+b$ à l'aide de trois opérateur NAND.

$$f = a+b = \text{NAND}(\text{NAND}(a,a), \text{NAND}(b,b))$$

le circuit correspondant est le suivant :



3.2.2 Opérateur NOR (Not OR)

La fonction suivante est le résultat de l'application de l'opérateur NOR sur les deux variables logiques a et b :

$$F(a, b) = \overline{a + b}$$

L'opérateur NOR est aussi un opérateur complet car il permet la réalisation des trois fonctions de base (ET, OU et NON).

4 Table de vérité (TV)

L'établissement de la table de vérité de F consiste à remplir d'abord toutes les combinaisons possibles d'entrées, cependant une fonction F contenant n variables d'entrées nécessite 2^n combinaisons (lignes) dans la table de vérité [6-7].

Par exemple la fonction $S=A \oplus B$ contient deux entrées A et B. La table de vérité suivante affiche les 4 (2^2) combinaisons possible ainsi que les valeurs de S associées à chaque combinaison.

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

4.1 Extraction des fonctions logiques à partir de la TV (FORMES CANONIQUES)

Une fonction peut être exprimée sous sa forme canonique à partir de sa propre table de vérité. On appelle forme canonique d'une fonction logique, la forme qui permette de localiser chaque ligne d'une table de vérité comportant un 1 ou 0. Il existe plusieurs formes canoniques : les plus utilisées sont la première et la deuxième forme.

4.1.1 Première forme canonique (forme disjonctive)

C'est la forme exprimée en somme de produits (SOP) (ou somme des **mintermes**). On dit aussi que c'est une disjonction de conjonctions. Cette forme est la forme la plus utilisée. Le principe est de localiser toutes lignes dont la variable de sortie vaut 1 (supposons qu'il y a p lignes). Pour chaque ligne, faire correspondre un produit de tous les variables d'entrée sous la forme normale si la variable d'entrée est à 1, sous la forme complément si la variable d'entrée est à 0. Faire ensuite la somme logique de ces p produits [2].

Exemple:

Trouver la 1^{ère} forme canonique S.O.P de la fonction à partir de la table de vérité suivante (tableau 3) :

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Tableau 3 : Extraction des formes canoniques à partir de la table de vérité

Dans ce cas la forme canonique de F est : $(\bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot c)$.

4.1.2 Deuxième forme canonique (conjonctive):

C'est la forme exprimée en produit de sommes (POS) (ou produit des **maxtermes**).

Le principe est de localiser toutes lignes dont la variable de sortie vaut 0 (supposons qu'il y a g lignes). Pour chaque ligne, faire correspondre une somme de tous les variables d'entrée sous la forme normale si la variable d'entrée est à 0, sous la forme complément si la variable d'entrée est à 1. Faire ensuite le produit logique de ces g produits [2].

Exemple :

Prenons la table de vérité ci-dessus, la 2^{ème} forme canonique POS de la fonction F est :

$$F_{SOP} = (a + b + \bar{c})(a + \bar{b} + c)(\bar{a} + \bar{b} + c)(\bar{a} + b + \bar{c})$$

5 Simplification des fonctions logiques.

La simplification d'une fonction logique consiste à réduire le nombre de termes de la fonction par application des postulats et des théorèmes de l'algèbre de Boole. Dans cette section, nous allons évoquer deux types de simplification : simplification algébrique et simplification par karnaugh.

5.1 Simplification Algébrique :

Il s'agit d'appliquer les postulats et les théorèmes de l'algèbre de Boole qui sont résumés ci-dessous :

Exemple :

Simplifier par la méthode algébrique les fonctions suivantes :

$$S = (x+y)(\bar{y} + z)(\bar{x} + z)$$

Solution

Appliquons la distribution du premier terme sur le deuxième

$$S = (x\bar{y} + xz + y\bar{y} + yz)(\bar{x} + z)$$

$y\bar{y} = 0$ (d'après le théorème de la complémentarité)

$$S = (x\bar{y} + xz + yz)(\bar{x} + z)$$

Appliquons les lois de distributivité, commutativité et de l'associativité des opérateurs logiques, nous obtenons :

$$S = x\bar{x}\bar{y} + x\bar{y}z + x\bar{x}z + xzz + \bar{x}yz + yzz$$

Sachant $zz=z$ (d'après l'idempotence)

$$S = x\bar{y}z + xz + \bar{x}yz + yz$$

D'après la distribution :

$$S = xz(\bar{y} + 1) + (\bar{x} + 1)yz$$

Sachant que $\bar{x} + 1 = 1$ (loi d'élément absorbant) alors $S = xz + yz$

5.2 Simplification par karnaugh :

La simplification par la méthode de karnaugh passe par les étapes suivantes :

- Création de la table de karnaugh (TK) selon le nombre de variables.
- Remplissage de la TK
- Repérer les cases adjacentes
- Simplification.

5.2.1 Création de la table de karnaugh (TK) :

La table de karnaugh est une réduction de la table de vérité dans le sens où pour n variables, la TK contient 2^n cases au lieu de 2^n lignes (cas de TV). Ces cases seront divisées en 2^p colonnes et 2^q lignes, avec $p+q=n$, et seront remplies selon le code binaire réfléchi (code Gray), pour lequel deux nombres adjacents ne diffèrent que d'un seul bit [8].

Table de karnaugh à 2 variables A et B

$N=2=1+1=p+q \rightarrow p=1$ et $q=1$ donc la TK contient 2 colonnes et 2 lignes

B \ A	0	1
0		
1		

Table de karnaugh à 3 variables A , B et C

$N=3=2+1=p+q \rightarrow p=2$ et $q=1$ donc la TK contient 4 colonnes et 2 lignes.

Nous remarquons que la combinaison $AB=11$ vient avant la combinaison 10 , car dans la TK, le passage d'une case vers une case adjacente, une seule variable qui change à la fois. Si on écrit les combinaisons des variables AB telles qu'on a l'habitude de travailler c'est-à-dire $00 / 01 / 10 / 11$, nous remarquons que dans le passage de la combinaison 01 à la combinaison 10 , les deux variables A et B changent à la fois, ce qui n'est pas permis dans la TK.

C \ AB	00	01	11	10
0				
1				

Table de karnaugh à 4 variables A , B , C et D

$N=4=2+2=p+q \rightarrow p=2$ et $q=2$ donc la TK contient 4 colonnes et 4 lignes.

CD \ AB	00	01	11	10
00				
01				
11				
10				

Remarque :

Une fonction qui contient 5 variables nécessite deux tables de karnaugh.

5.2.2 Remplissage de la table de karnaugh :

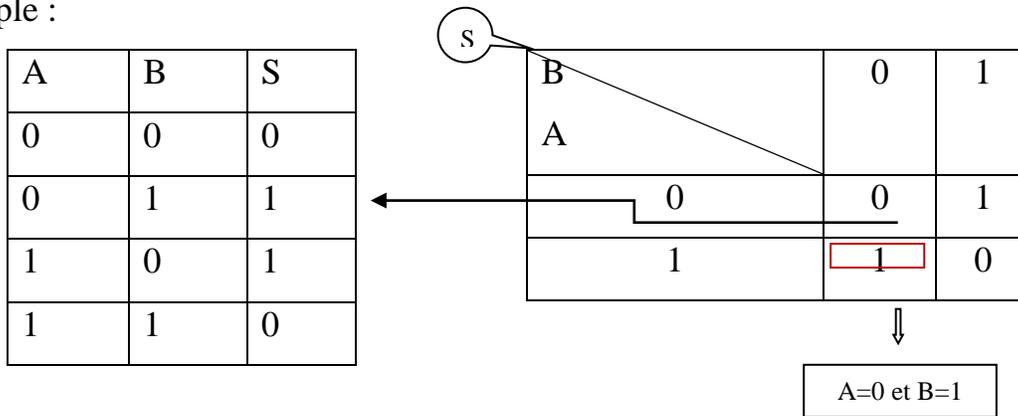
Il existe plusieurs manières pour remplir une table de karnaugh :

- Remplissage à partir d'une table de vérité
- Remplissage à partir d'une forme SOP
- Remplissage à partir d'une forme POS

Remplissage à partir d'une table de vérité :

Chaque case dans la TK correspond à une ligne dans la table de vérité :

Par exemple :



Remplissage à partir d’une forme SOP :

Dans ce cas pour chaque minterme, la valeur de la fonction est égale à 1.

Si l’opérateur de négation est appliqué sur une variable (par exemple \bar{a}) alors cette dernière est égale à zero (a=0) sinon elle est égale à un [9-10].

Soit la forme SOP suivante :

$$F_{sop} = (\bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot c)$$

$$(\bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot c)$$



$$(0.0.0 + 0.1.1 + 1.0.0 + 1.1.1)$$



Remarque : pour les autres combinaisons la fonction F est égale à zero.

La TK de la fonction F_{sop} à 03 variables a,b et c sera remplie de la manière suivante :

C \ AB	00	01	11	10
0	1	0	0	1
1	0	1	1	0

Remplissage à partir d'une forme POS :

Dans ce cas pour chaque maxterme, la valeur de la fonction est égale à 0.

Si l'opérateur de négation est appliqué sur une variable (par exemple \bar{a}) alors cette dernière est égale à un ($a=1$) sinon elle est égale à zero.

Soit la forme POS suivante :

$$\begin{array}{cccc}
 F_{\text{POS}} = (a + b + \bar{c})(a + \bar{b} + c)(\bar{a} + \bar{b} + c)(\bar{a} + b + \bar{c}) & & & \\
 \Downarrow & \Downarrow & \Downarrow & \Downarrow \\
 (0 + 0 + 1)(0 + 1 + 0)(1 + 1 + 0)(1 + 0 + 1) & & & \\
 \Downarrow & \Downarrow & \Downarrow & \Downarrow \\
 F=0 & F=0 & F=0 & F=0
 \end{array}$$

Remarque : pour les autres combinaisons la fonction F est égale à un.

C \ AB	00	01	11	10
0	1	0	0	1
1	0	1	1	0

5.2.3 *Notion des cases adjacentes :*

deux cases sont adjacentes si elles diffèrent d'une seule variable. Dans les tableaux ci-dessous, les cases X représentent les cases adjacentes associées à la case bleue.

C \ AB	00	01	11	10
0		X		
1	X		X	

CD \ AB	00	01	11	10
00	X			
01				
11	X			
10		X		X

Remarque : si une fonction F contient N variables alors chaque case dans la TK possède N cases adjacentes.

5.2.4 Simplification par karnaugh :

Il s'agit de faire le regroupement des cases adjacentes :

Si on fait des regroupements des « uns », on aura une fonction simplifiée sous la forme SOP. Dans le cas où on fait des regroupements des « zeros », on aura une fonction simplifiée sous la forme POS.

Exemple :

Donner la fonction simplifiée de S sous la forme SOP. la fonction S est donnée par la TK suivante :

S \ C \ AB	00	01	11	10
0	1	0	0	1
1	0	1	0	1

Diagram illustrating the Karnaugh map for function S with groupings:

- Group 1 (Red): A horizontal group of two 1s in the top row (00 and 10).
- Group 2 (Blue): A vertical group of two 1s in the rightmost column (01 and 10).
- Group 3 (Black): A single 1 in the bottom row, middle column (01).

Dans ce cas, il s'agit de faire des regroupements des cases adjacentes contenant des « uns ».

- Chaque « un » doit être casé dans un groupe.
- Un même « un » peut être casé dans plusieurs groupes.

- Un regroupement peut contenir 1, 2, 4, .. jusqu'à 2^n cases adjacentes.

Nous remarquons dans l'exemple précédent que :

- Aucune case adjacente à la case 011 (A=0, B=1 et C=1) ne contient la valeur « un », ce qui donne un regroupement d'une seule case. Pour les autres cases, ça sera un regroupement de deux.
- Dans la fonction simplifiée, on garde uniquement les variables qui ne changent pas d'état. N'oublions pas que dans la forme SOP, si la variable A=0 alors j'écris \bar{A} dans la fonction sinon j'écris A. Par exemple pour le groupe 1, nous obtenons le terme $\bar{B} \bar{C}$, dans ce cas la variable A est éliminée car elle change d'état. Pour le groupe 2, nous obtenons $A\bar{B}$, ici la variable C est éliminée et pour le groupe 3 aucune variable n'est éliminée, nous obtenons donc $\bar{A}BC$.
- La fonction simplifiée sous la forme SOP (somme des produits) consiste à mettre la somme entre les groupes (dans cet exemple il y a 03 groupes) et le produit entre les variables. La fonction f simplifiée devient :

$$F = \bar{B}\bar{C} + A\bar{B} + \bar{A}BC$$

Remarque :

Une fonction simplifiée sous la forme POS (produit de somme) consiste à faire le produit entre les groupes et la somme entre les variables. Il s'agit de faire des regroupements des zeros, si la variable A=1 alors j'écris \bar{A} dans la fonction sinon j'écris A.

Le but de la simplification par karnaugh est :

- Avoir un minimum de groupes
- Avoir un maximum des uns (ou zeros) dans chaque groupe
- Une case peut appartenir à plusieurs groupes.

5.2.5 Simplification par karnaugh à partir d'une table incomplète:

Il arrive dans, quelques problèmes, que la fonction logique ne soit pas définie pour quelques combinaisons des valeurs des entrées. La valeur de la fonction F dans ce cas est notée par X. La table de karnaugh résultante est dit table incomplète. Nous pouvons affecté n'importe quelle valeur (0 ou 1) aux valeurs X de la table incomplète de manière à regrouper un maximum d'éléments dans chaque groupe formé.

Exemple : Donner la fonction simplifiée de la table de karnaugh suivante :

F	AB	00	01	11	10
C					
0		x	0	x	0
1		0	1	x	x

$$F = B.C$$

6 Conclusion :

Dans ce chapitre nous avons donné un aperçu général sur l'algèbre de Boole, nous avons vu comment simplifier une fonction logique algébriquement et par la méthode de karnaugh. Cette dernière est expliquée d'une manière simple et détaillée afin que nos étudiant

7 Exercices

Exercice 1

Simplifiez par la méthode algébrique les fonctions suivantes :

$$F0 = x(\bar{x} + y)$$

$$F1 = x + \bar{x}.y$$

$$F2 = x.y + \bar{x}$$

$$F3 = (x+y). (x+z). (y+z)$$

$$F4 = \bar{x}.\bar{y} + \bar{y}$$

$$F5 = \overline{\overline{x+y}} + x.y.(\overline{z+y})$$

$$F6 = \overline{\overline{x+y+z}.t} + \overline{\overline{x+y}.z}$$

$$F7 = x.(x.\bar{y}(y+x.t)) + \bar{x} + \bar{t}$$

$$F8 = \overline{\overline{x+y+z}} + \overline{\overline{x+y}.z}$$

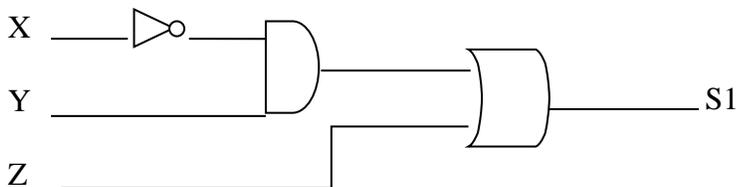
$$F9 = (\bar{x} + \bar{y}). (x+y)$$

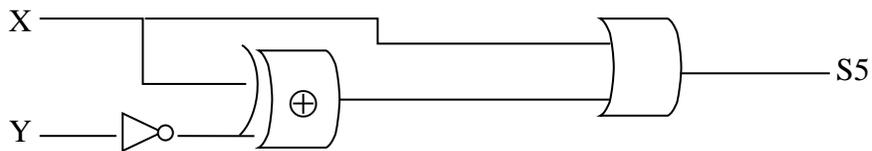
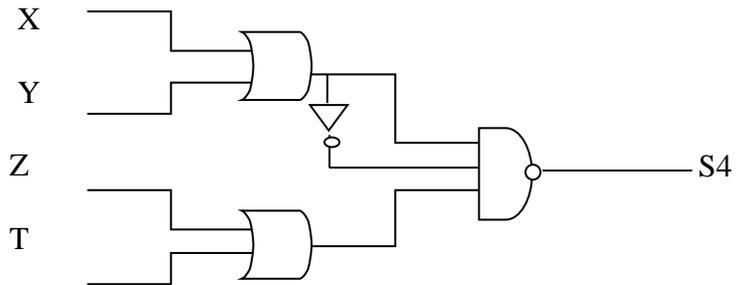
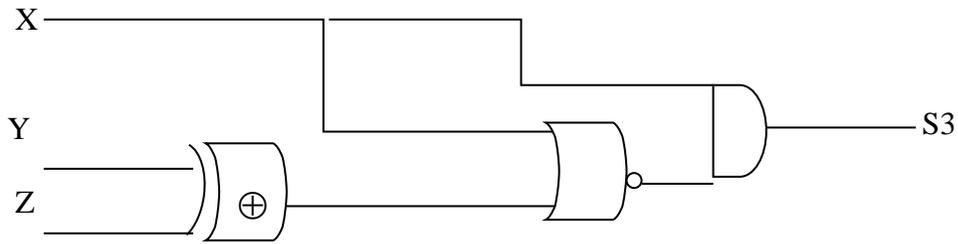
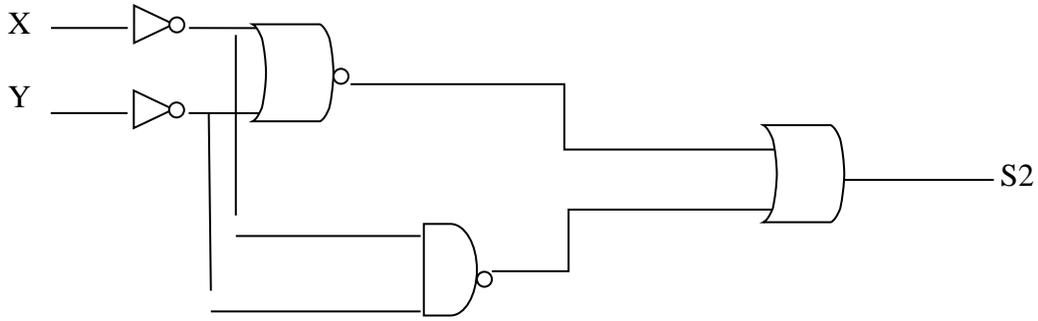
$$F10 = (\bar{x} + y). (x + \bar{y})$$

Exercice 2

On suppose que X, Y, Z et T sont des variables booléennes, donner pour chaque circuit ci-dessous :

1. Sa fonction algébrique Si
2. Sa table de vérité.
3. Sa fonction simplifiée.





Exercice 3 :

Donner la 1^{ère} forme canonique (SOP) puis la 2^{ème} forme canonique (POS) de chaque fonction .

$$F1 = x + y + \bar{x} \cdot (y+z)$$

$$F2 = ((x + y) \cdot (x + \bar{y}) + y) \cdot (\bar{x} + y)$$

$$F3 = (x + y) \cdot (\bar{y} + z) \cdot (\bar{x} + z)$$

$$F4 = \overline{x + y} + (x \oplus \bar{y})$$

$$F5 = (x + y) \cdot (\bar{y} + z) + \bar{t} (x + y)$$

Exercice 4 :

Simplifier les fonctions suivantes (F1,...F12) par la méthode de karnaugh. Mettre les deux premières fonctions de chaque tableau sous la forme simplifié en choisissant les boucles de « 1 » et les deux dernières fonctions en choisissant les boucles de « 0 » .

1)

A	B	F1	F2	F3	F4
0	0	1	1	0	0
0	1	0	1	1	1
1	0	1	0	1	0
1	1	0	1	0	0

2)

A	B	C	F5	F6	F7	F8
0	0	0	1	0	1	0
0	0	1	0	1	1	0
0	1	0	1	1	1	1
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	1	1	0	0
1	1	0	1	1	0	1
1	1	1	1	1	1	1

3)

A	B	C	D	F9	F10	F11	F12
0	0	0	0	0	1	0	1
0	0	0	1	0	1	1	1
0	0	1	0	0	1	0	1
0	0	1	1	1	1	1	1
0	1	0	0	1	1	1	0
0	1	0	1	0	0	1	0
0	1	1	0	1	0	0	1
0	1	1	1	0	1	1	0
1	0	0	0	1	0	0	1
1	0	0	1	1	1	0	0
1	0	1	0	0	1	0	0
1	0	1	1	0	0	0	0
1	1	0	0	1	1	0	1
1	1	0	1	0	1	1	0
1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0

Exercice 5 :

Simplifier les tables de karnaugh suivantes avec des valeurs incomplètes.

I1I2 \ I3	00	01	11	10
0	0	x	1	0
1	x	1	1	1

I1I2 \ I3	00	01	11	10
0	x	x	0	1
1	x	1	1	1

AB \ CD	00	01	11	10
00	0	x	0	x
01	0	0	1	0
11	x	x	x	1
10	0	0	1	0

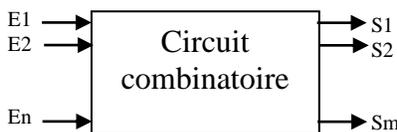
AB \ CD	00	01	11	10
00	0	0	1	1
01	x	x	0	1
11	x	1	0	0
10	0	0	x	1

Chapitre 2

Les circuits combinatoires

1 Introduction

Les circuits combinatoires souvent vus comme des boîtes noires, sont des circuits numériques associant pour chaque combinaison d'entrées $\{E1, E2, \dots, E_n\}$ une seule et même combinaison de sortie $\{S1, S2, \dots, S_m\}$. Dans un circuit combinatoire, la sortie S_i s'écrit en fonction des entrées de la manière suivante: $S_i = F(E1, E2, \dots, E_n)$, tel que F est la fonction décrivant le comportement de ce circuit [11].



2 Etapes de conception d'un circuit combinatoire:

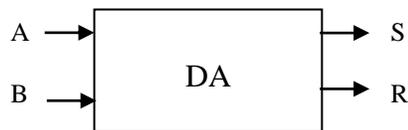
La réalisation d'un circuit combinatoire passe par les étapes suivantes :

- Etablissement de la table de vérité.
- Extraction des fonctions logiques exprimant les sorties en fonction des entrées
- Simplification des fonctions logiques.
- Réalisation du circuits logique (ou le logigramme)

3 Etude de quelques circuits combinatoires usuels :

3.1 Le demi-additionneur (DA):

Le DA est un circuit combinatoire qui permet de réaliser la somme **sur un bit** de deux nombre A et B. ce circuit contient donc deux entrées A et B et génère deux sorties : S qui représente la somme de A et B sur un bit et R qui représente la retenue (carry en anglais). Il faut noter que le DA ne tient pas en compte la retenue précédente. La boîte noire du DA est donnée par le schéma suivant :



Voici le scénario qu'on peut avoir en additionnant A avec B :

A=0	A=0	A=1	A=1
B=0	B=1	B=0	B=1
S=0	S=1	S=1	S=0
R=0	R=0	R=0	R=1

La table de vérité du DA contient donc $2^{\text{nombre_de_variables_d'entrées}} = 2^2 = 4$ lignes

A	B	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Question : Ecrire S puis R sous la forme SOP puis donner le schéma du DA.

Réponse : $S = \bar{A}.B + A.\bar{B} = A \oplus B$ et $R = A.B$

Le circuit combinatoire du DA est donc donnée par le schéma suivant :

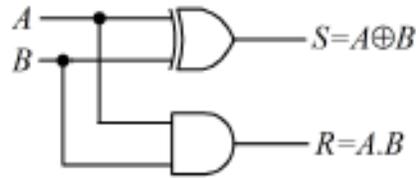


Figure 1: Logigramme d'un demi-Additionneur.

3.2 L'additionneur complet :

L'additionneur complet (AC) est un circuit combinatoire qui permet de réaliser la somme **sur un bit** de deux nombre A et B tout en tenant en compte la retenue précédente (R_{i-1}) [12]. Ce circuit contient donc trois entrées A, B et R_{i-1} et génère deux sorties : S qui représente la somme de A et B sur un bit et R_i qui représente la retenue. La boîte noire du AC est donnée par le schéma suivant :

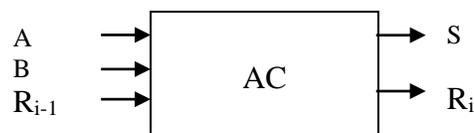


Figure 2 : Boîte noire d'un additionneur complet.

La figure suivante montre un exemple d'addition de deux nombres de 4 bits ainsi que les reports (retenues) transférés aux étages suivants :

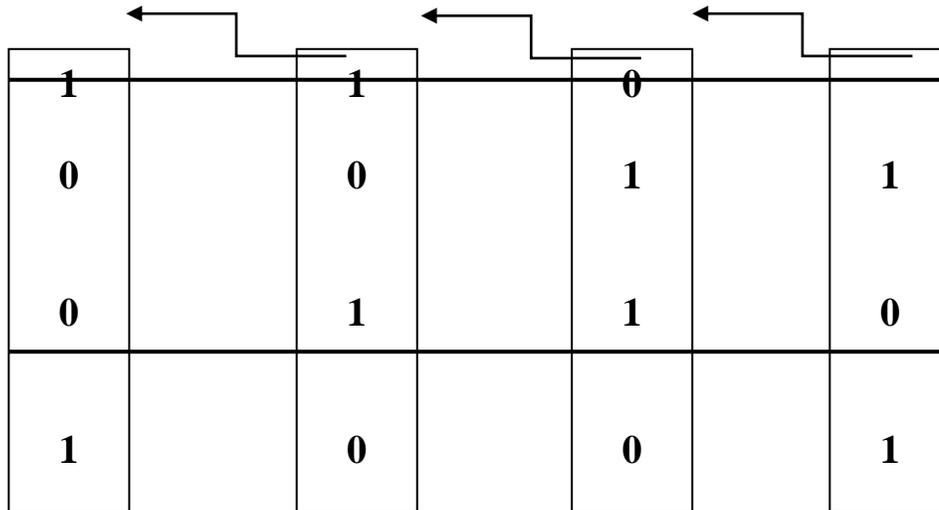


Figure 3: schéma d'addition de deux nombres de 4 bits.

La table de vérité du AC contient donc $2^{\text{nombre_de_variables_d'entrées}} = 2^3 = 8$ lignes

A	B	R_{i-1}	S	R_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Extraction des formes SOP de S et R_i

$$\begin{aligned}
 S &= \bar{A} \cdot \bar{B} \cdot R_{i-1} + \bar{A} \cdot B \cdot \overline{R_{i-1}} + A \cdot \bar{B} \cdot \overline{R_{i-1}} + A \cdot B \cdot R_{i-1} = (\bar{A} \cdot \bar{B} + A \cdot B) \cdot R_{i-1} + (\bar{A} \cdot B + A \cdot \bar{B}) \cdot \overline{R_{i-1}} \\
 &= (\overline{A \oplus B}) \cdot R_{i-1} + (A \oplus B) \cdot \overline{R_{i-1}} = A \oplus B \oplus R_{i-1}
 \end{aligned}$$

$$R_i = \bar{A}.B.R_{i-1} + A.\bar{B}.R_{i-1} + A.B.\overline{R_{i-1}} + A.B.R_{i-1} = (\bar{A}.B + A.\bar{B}) R_{i-1} + A.B(\overline{R_{i-1}} + R_{i-1})$$

$$\text{Donc } R_i = (A \oplus B). R_{i-1} + A.B$$

Le circuit combinatoire du AC est donnée par le schéma suivant :

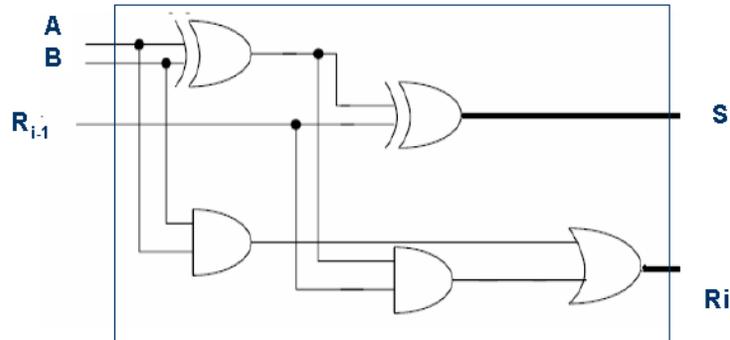


Figure 4: Logigramme d'un additionneur complet.

Réalisation d'un additionneur complet en utilisant des demi-additionneurs.

Nous avons déjà vu dans les sections précédentes que les sorties d'un AC s'écrivent :

$$\left. \begin{aligned} S &= A \oplus B \oplus R_{i-1} \\ R_i &= (A \oplus B). R_{i-1} + A.B \end{aligned} \right\} \textcircled{1}$$

Et les sorties d'un demi-additionneur s'écrivent :

$$S_{DA} = A \oplus B$$

$$R_{DA} = A.B$$

En remplaçant les sorties du DA dans (1), les sorties du AC deviennent :

$$S = S_{DA} \oplus R_{i-1}$$

$$R_i = S_{DA}. R_{i-1} + R_{DA}$$

L'additionneur complet est réalisé donc à partir de deux demi-additionneurs : le premier réalise l'addition des deux nombres A et B et le deuxième réalise l'addition des deux nombre $A \oplus B$ et R_{i-1} .

Voici le schéma d'un AC en utilisant des DA

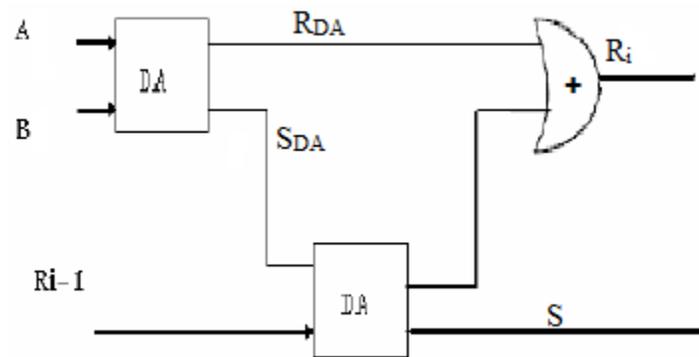


Figure 5: Additionneur complet réalisé à partir des demi-additionneur

3.3 Le Demi-soustracteur (DS) :

Le DS est circuit combinatoire qui permet de faire la soustraction sur un bit de deux nombres A et B. le DS génère deux sorties D et R, D qui représente le résultat de la soustraction et R la retenue.

Remarque : le DS ne tient pas en compte de la retenue précédente.

La table de vérité du DS est donnée comme suit :

A	B	D	R
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

La forme SOP de la fonction D est donnée par :

$$D = \bar{A}.B + A.\bar{B} = A \oplus B$$

Et la forme SOP de la fonction R est donnée par : $R = \bar{A} \cdot B$

Le circuit combinatoire du DS est donc donné par le schéma suivant :

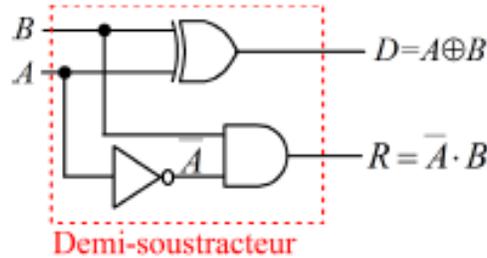


Figure 6: Logigramme d'un demi-soustracteur.

3.4 Le soustracteur complet :

Le soustracteur complet (SC) est un circuit combinatoire qui permet de réaliser la soustraction **sur un bit** de deux nombre A et B tout en tenant en compte la retenue précédente (R_{i-1}). Ce circuit contient donc trois entrées A, B et R_{i-1} et génère deux sorties : S qui représente le résultat de la soustraction de A et B sur un bit et R_i qui représente la retenue. La table de vérité du SC est donnée par le tableau suivant :

A	B	R_{i-1}	D	R_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D = \bar{A} \cdot \bar{B} \cdot R_{i-1} + \bar{A} \cdot B \cdot \overline{R_{i-1}} + A \cdot \bar{B} \cdot \overline{R_{i-1}} + A \cdot B \cdot R_{i-1} = (\bar{A} \cdot \bar{B} + A \cdot B) \cdot R_{i-1} + (\bar{A} \cdot B + A \cdot \bar{B}) \cdot \overline{R_{i-1}}$$

$$= (\bar{A} \oplus \bar{B}) \cdot R_{i-1} + (A \oplus B) \cdot \overline{R_{i-1}} = A \oplus B \oplus R_{i-1}$$

$$R_i = \bar{A}. \bar{B}.R_{i-1} + \bar{A}.B \overline{R_{i-1}} + \bar{A}.B.R_{i-1} + A.B.R_{i-1} = (\bar{A}. \bar{B} + A.B) R_{i-1} + \bar{A}.B(\overline{R_{i-1}} + R_{i-1})$$

Donc $R_i = (\bar{A} \oplus B). R_{i-1} + \bar{A}.B.$

Remarque : le soustracteur complet sur un bit est réalisé à partir de deux demi-soustracteur comme le montre la figure ci-dessous.

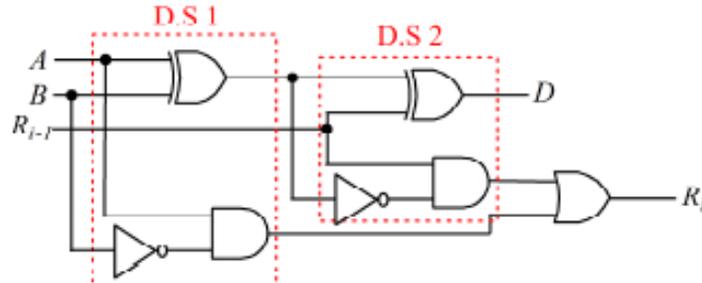


Figure 7: Logigramme d'un soustracteur complet 1 bit.

3.5 Le comparateur :

Le comparateur 1 bit est un circuit combinatoire qui permet de comparer deux nombres A et B de 1 bit et génère trois sorties : Ss, Se, Si. Son fonctionnement est le suivant :

- Ss=1 si A > B,
- Se=1 si A = B,
- Si=1 si A < B.

La boîte noire du comparateur 1 bit est donnée par le schéma suivant :

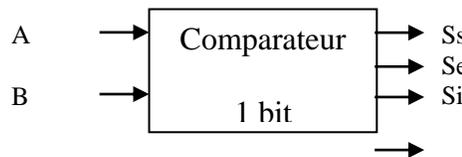


Figure 8: Boîte noire d'un comparateur 1 bit.

La table de vérité du comparateur 1 bit contient donc $2^{\text{nombre_de_variables_d'entrées}} = 2^2 = 4$ lignes

A	B	Ss	Se	Si
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

Question : Ecrire Ss, Se puis Si sous la forme SOP puis donner le schéma du comparateur 1 bit.

Réponse :

$$Ss = A \cdot \bar{B}$$

$$Se = \bar{A} \cdot \bar{B} + A \cdot B = \overline{A \oplus B}$$

$$Si = \bar{A} \cdot B$$

Le circuit combinatoire du comparateur 1 bit est donc donnée par le schéma suivant :

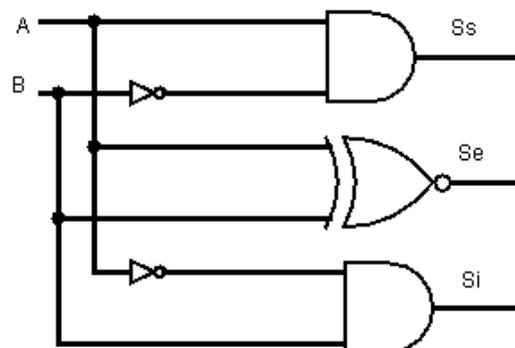


Figure 9: Circuit logique d'un comparateur 1 bit.

3.6 Le codeur (ou encodeur) :

Le codeur est un circuit combinatoire de N entrées représentant l'objet à coder (par exemple un nombre décimal, un chiffre, un son, une image, etc..) et M sorties représentant le code de l'objet. Le principe de fonctionnement d'un codeur est le suivant : lorsqu'une entrée est

activée, les sorties affichent la valeur correspondant au numéro de l'entrée dans le code binaire choisi. Dans un codeur, une seule entrée doit être active à la fois.

Exemple : le codeur décimal - BCD qui permet de coder un chiffre décimal en son équivalent BCD [13]. La boîte noire du codeur décimal - BCD est représentée par le schéma suivant :

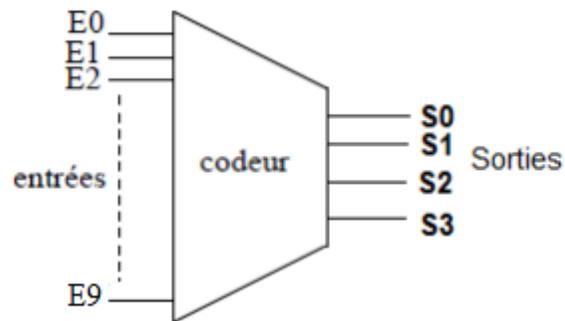


Figure 10: Schéma du codeur decimal-BCD.

La table du codeur est donnée en dessous.

Les équations logiques des sorties sont données comme suit :

$$S_0 = E_1 + E_3 + E_5 + E_7 + E_9$$

$$S_1 = E_2 + E_3 + E_6 + E_7$$

$$S_2 = E_4 + E_5 + E_6 + E_7$$

$$S_3 = E_8 + E_9$$

Entrées	Sorties			
	S ₃	S ₂	S ₁	S ₀
E0	0	0	0	0
E1	0	0	0	1
E2	0	0	1	0
E3	0	0	1	1
E4	0	1	0	0
E5	0	1	0	1
E6	0	1	1	0
E7	0	1	1	1
E8	1	0	0	0
E9	1	0	0	1

Le circuit logique du codeur est donnée par le schéma suivant

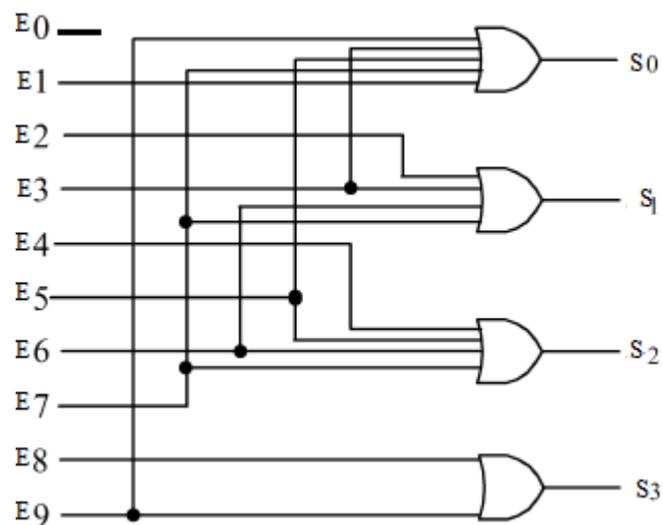


Figure 11: Circuit logique du codeur decimal-BCD

3.7 Le décodeur :

Le décodeur joue le rôle inverse du codeur. Étant donné un code sur N bits (N entrées), le décodeur permet de décoder ce code en signalant sa présence en mettant un signal en sortie à 1.

Exemple :

Le décodeur 3 vers 8, qui permet de décoder un nombre binaire écrit sur 03 bits en son équivalent décimal [14-15]. La table de vérité de ce décodeur est la suivante :

Entrées			Sorties							
X ₀	X ₁	X ₂	S ₀	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆	S ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Tableau 4 : La table de vérité du décodeur 3 vers 8

Les équations logiques des sorties S₀...S₇ sont les suivantes :

$$S_0 = \overline{X_0} \cdot \overline{X_1} \cdot \overline{X_2}$$

$$S_1 = \overline{X_0} \cdot \overline{X_1} \cdot X_2$$

$$S_2 = \overline{X_0} \cdot X_1 \cdot \overline{X_2}$$

$$S_3 = \overline{X_0} \cdot X_1 \cdot X_2$$

$$S_4 = X_0 \cdot \overline{X_1} \cdot \overline{X_2}$$

$$S_5 = X_0 \cdot \overline{X_1} \cdot X_2$$

$$S_6 = X_0 \cdot X_1 \cdot \overline{X_2}$$

$$S_7 = X_0 \cdot X_1 \cdot X_2$$

Finalement le schéma logique du décodeur 3 vers 8 est représentée ci-dessous :

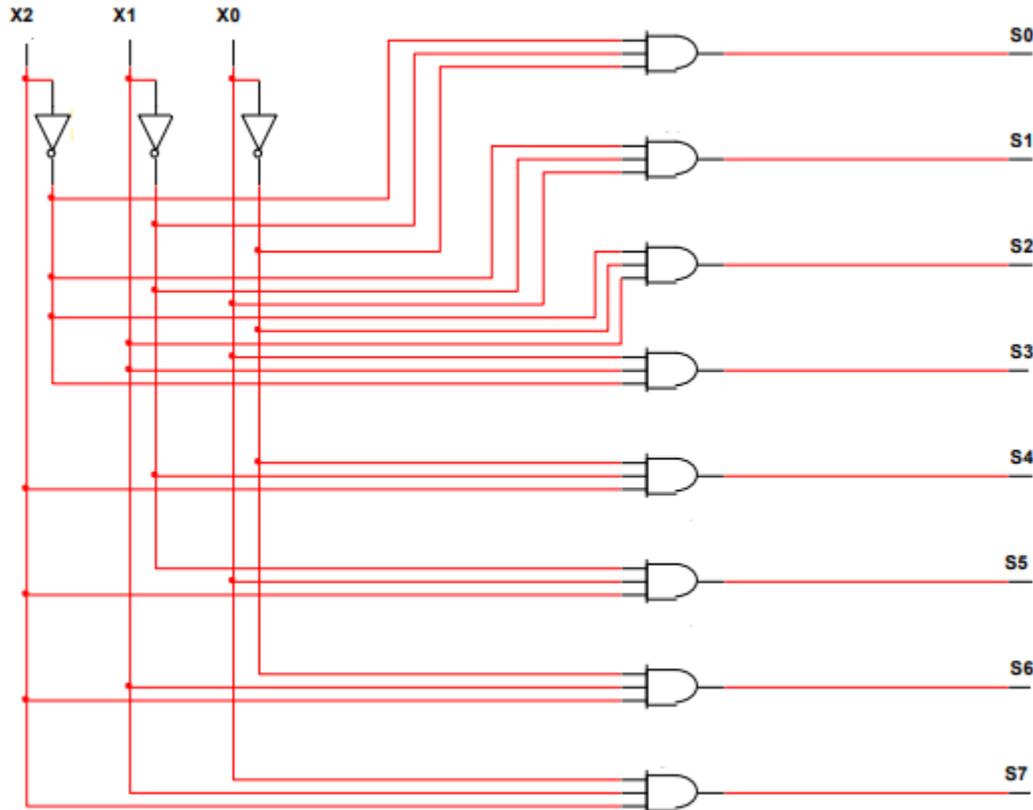


Figure 12: Circuit logique du décodeur 3 vers 8.

3.8 Le Transcodeur :

Le transcodeur est un circuit combinatoire permettant de transformer un code en un autre code équivalent. Le transcodeur contient N entrées qui représentent l'objet à transformer (par exemple un nombre binaire, BCD, Gray, etc..) et M sorties qui représentent le code équivalent. Exemple de transcodeur : transcodeur Binaire – Gray, transcodeur BCD - XS-3, etc ..

Exemple : faire le circuit combinatoire du transcodeur BCD - XS-3.

Ce transcodeur contient 04 entrées qui correspondent au code BCD (ABCD) et 04 sorties correspondent au code XS-3 (XYZT). Remarque : le code XS-3 est appelé aussi code BCD+3. Dressons la table de vérité de ce transcodeur.

Décimale	Entrées (Code BCD)				Sorties (Code XS-3)			
	A	B	C	D	X	Y	Z	T
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

Les 06 combinaisons de 1010 jusqu'à 1111 sont dites interdites. Dressons les 4 tables de karnaugh correspondantes aux 04 sorties X, Y, Z et T.

AB \ CD	00	01	11	10
00	0	0	x	1
01	0	1	x	1
11	0	1	x	X
10	0	1	x	X

$$X = A + B \cdot C + B \cdot D = A + B \cdot (C + D)$$

$$\overline{C} \cdot \overline{D}$$

AB \ CD	00	01	11	10
00	0	1	x	0
01	1	0	x	1
11	1	0	x	X
10	1	0	x	X

$$Y = \overline{B} \cdot C + \overline{B} \cdot D + B$$

$$= \overline{B} \cdot (C + D) + B \cdot (\overline{C + D})$$

$$= B \oplus (C + D)$$

AB \ CD	00	01	11	10
00	1	1	x	1
01	0	0	x	0
11	0	0	x	X
10	1	1	x	X

$$Z = \overline{D}$$

AB \ CD	00	01	11	10
00	1	1	x	1
01	0	0	x	0
11	1	1	x	X
10	0	0	x	X

$$T = \overline{C} \cdot \overline{D} + C \cdot D = \overline{C \oplus D}$$

Le circuit logique du transcodeur BCD – XS-3

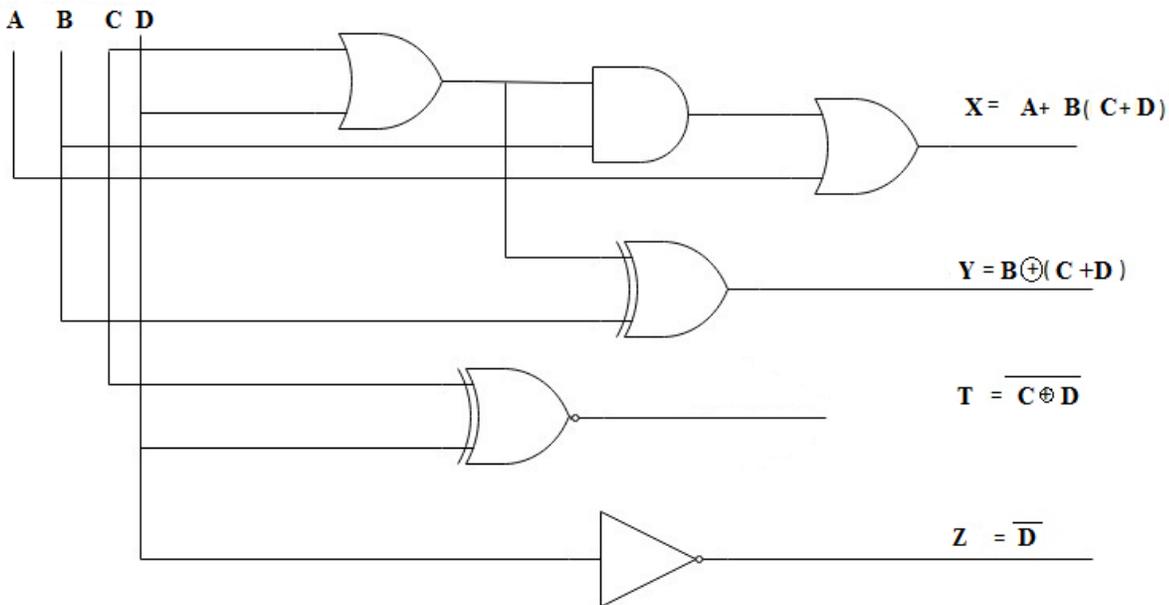


Figure 13: Circuit logique d'un transcodeur BCD – XS-3.

3.9 Le multiplexeur :

Un multiplexeur est un circuit combinatoire contenant N entrées d'informations ou de données (E_0, E_1, \dots, E_{N-1}) et n entrées de selection ou entrées d'adresse ou d'aiguillage (C_0, C_1, \dots, C_{n-1}) et une seule sortie. N s'écrit sous forme de 2^n . La sortie d'un multiplexeur est

égale à une des entrées, le multiplexeur réalise donc un aiguillage de l'une de ses entrées vers une sortie unique selon les bits d'adresse.

Par exemple si on a 4 ($=2^2$) entrées d'informations, nous allons avoir 2 entrées d'adresse. Pour la combinaison (00) des entrées d'adresse, l'entrée E0 (la première entrée) qui sera aiguillée vers la sortie. Et pour la combinaison (01) l'entrée E1 qui sera aiguillée vers la sortie et ainsi de suite.

Le schéma du multiplexeur est le suivant :

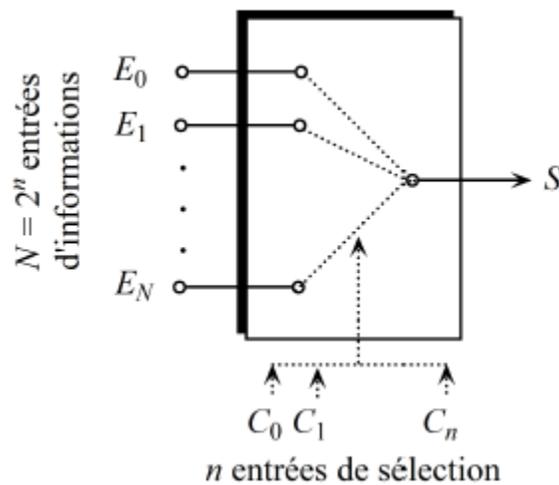


Figure 14: Schéma d'un multiplexeur.

Exemple : Multiplexeur 4 vers 1.

Ce multiplexeur contient 4 entrées d'informations ($4=2^2$), représentées par E₀, E₁, E₂ et E₃ et contient donc 2 entrées de sélection (C₀ et C₁) et une seule sortie (S). La table de vérité de ce multiplexeur est la suivante :

C0	C1	S
0	0	E0
0	1	E1
1	0	E2
1	1	E3

L'équation logique de S est la suivante :

$$S = \overline{C_0} \cdot \overline{C_1} \cdot E_0 + C_0 \cdot \overline{C_1} \cdot E_1 + \overline{C_0} \cdot C_1 \cdot E_2 + C_0 \cdot C_1 \cdot E_3$$

Son logigramme est donnée par le schéma suivant :

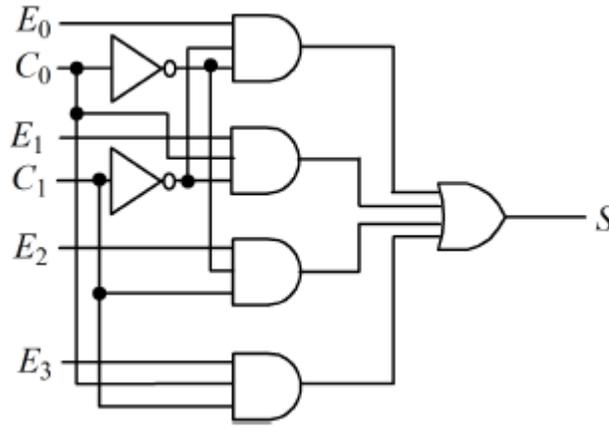


Figure 15: Circuit logique d'un multiplexeur 4 vers 1.

3.10 Le démultiplexeur

Le démultiplexeur est un circuit combinatoire à une seule entrée (E) et N sorties (S_0, S_1, \dots, S_{N-1}) tel que $N=2^n$ et n représente le nombre des entrées de sélection. Le démultiplexeur joue le rôle inverse d'un multiplexeur. Il s'agit d'aiguiller l'entrée E vers une sortie S_i sélectionnée par les entrées de sélection.

Exemple : demultiplexeur 1 vers 4

Ce circuit contient 1 entrée (E) et 4 ($=2^2$) sorties (S_0, S_1, S_2 et S_3) et 2 entrées de sélection (C_0 et C_1).

La table de vérité du démultiplexeur est la suivante

C0	C1	S ₀	S ₁	S ₂	S ₃
0	0	E	0	0	0
0	1	0	E	0	0
1	0	0	0	E	0
1	1	0	0	0	E

Tableau 5 : La table de vérité du démultiplexeur 1 vers 4

Les équations logiques des sorties sont données par :

$$S_0 = \overline{C_0} \cdot \overline{C_1} \cdot E$$

$$S_1 = \overline{C_0} \cdot C_1 \cdot E$$

$$S_2 = C_0 \cdot \overline{C_1} \cdot E$$

$$S_3 = C_0 \cdot C_1 \cdot E$$

Le logigramme du démultiplexeur 1 vers 4 est le suivant :

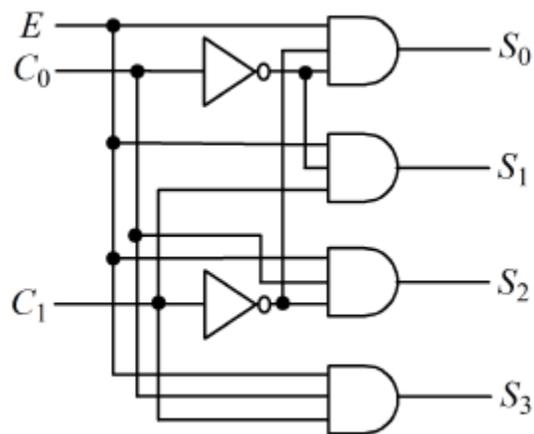


Figure 16: Circuit logique d'un démultiplexeur 1 vers 4.

4 Conclusion

Dans ce chapitre, nous avons essayé de donner un aperçu général sur les circuits combinatoires. Nous avons aussi donner quelques exemples des circuits combinatoires connus, chaque circuit est décrit par sa table de vérité, ses équations logiques en exprimant les sorties en fonction des entrées ainsi que son logigramme. Dans le chapitre suivant nous évoquerons les circuits séquentiels, nous allons voir la différence entre ces circuits et les circuits combinatoires.

5 Exercices

Exercice 1:

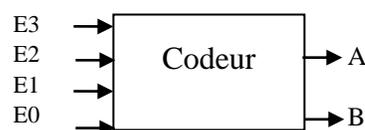
On donne ci-dessous la table de transcodage suivante :

Entrées			Sorties		
A3	A2	A1	B3	B2	B1
0	0	0	0	1	1
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	0	1
1	1	0	0	1	0
1	1	1	1	0	0

1. Donner la deuxième forme canonique (POS) de la fonction B1 .
2. Donner la première forme canonique (SOP) des fonctions B2 et B3 .
3. Simplifier les sorties du transcodeur (B3, B2, B1) par la méthode de karnaugh.
4. Donner les circuits logiques des fonctions simplifiées à l'aide des portes NAND.

Exercice 2:

Soit un circuit combinatoire à 4 lignes d'entrée et 2 lignes de sorties comme le montre la figure ci-dessous : Le fonctionnement est le suivant :



- Lorsqu'une seule ligne d'entrée, parmi E0, E1, E2, E3 se trouve au niveau haut (=1), son numéro est codé en binaire sur les sorties (BA),
 - Si plusieurs lignes sont simultanément au niveau haut (=1), on code le numéro le plus élevé,
 - Si toutes les lignes d'entrée sont au niveau bas (=0), on code (BA) = (00).
- 1) Dresser la table de vérité du codeur.
 - 2) Etablir les équations logiques des sorties A, B en fonction des entrées de E0...E3
 - 3) Représenter le schéma logique du codeur.

Exercice 3 :

Trois interrupteur I1, I2 et I3 commandent le démarrage de deux moteurs M1 et M2 selon les conditions suivantes

Le moteur M1 ne doit démarrer que si au moins deux interrupteurs sont activés (=1),
Dès qu'un ou plusieurs interrupteurs sont activés, le moteur M2 doit démarrer.

- 1) Donner la table de vérité correspondante,
- 2) Déterminer les équations simplifiées des sorties M1 et M2 par la méthode de karnaugh.
- 3) Faire le logigramme des sorties M1 et M2 simplifiées.

Exercice 4

Soient a,b,c, d des variables logiques, on considère les fonctions logiques F et G définies par :

$F = 1$ si et seulement si $a+b \leq c+d$

$G = 1$ si et seulement si abcd est strictement inférieur à 2(10).

Dresser la table de vérité de ces fonctions.

Donner les formes canoniques de F et G.

Simplifier les fonctions de sorties F et G par la méthode de karnaugh.

Exercice 5

Soit un circuit combinatoire à 04 variables d'entrées A,B,C,D et 03 sorties S1, S2 et S3 telles que :

S1 est vraie si (A est vraie) et (deux autres variables au moins sont vraies),

S2 est vraie si (seule A est fausse) ou (A est vraie et deux autres variables au moins sont fausses).

S3 est vraie si (A est fausse et une autre variable au moins est fausse).

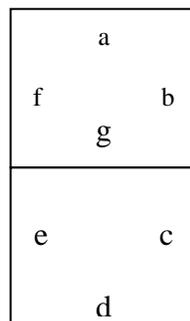
1) Etablir la table de vérité de ce circuit (Entrées + sorties).

2) Ecrire les équations simplifiées (par tableau de Karnaugh) des sorties S1, S2 et S3.

3) Dessiner le logigramme des sorties simplifiées S1, S2 et S3.

Exercice 6 : Afficheur 7 segments

Donner la table de vérité d'un décodeur BCD - 7 segments qui permet d'afficher les 10 chiffres décimaux à l'aide de 7 segments, notés de a à g. un segment prends la valeur 0 s'il est éteint ou 1 s'il est allumé. Les 7 segments sont disposés de la manière suivante :



Lorsqu'on a la combinaison 0 0 0 0 en entrée, le zero décimal s'affiche en activant les sorties a b c d e et f. et lorsqu'on a la combinaison 0 0 0 1 en entrée, le un décimal s'affiche en activant les sorties b et c et ainsi de suite.

Exercice 7 :

Donner la fonction algébrique d'un multiplexeur 8 vers 1.

Chapitre 3

La logique séquentielle.

1 Introduction

Contrairement à la logique combinatoire où les sorties du circuit dépendent seulement des entrées, la logique séquentielle permet de mémoriser les sorties précédentes du circuit qui vont intervenir sur les valeurs des sorties actuelles. Autrement dit, l'état de sortie du circuit séquentiel à un instant t dépend aux valeurs d'entrées à l'instant t ainsi que les valeurs de sorties aux instants antérieurs (soit $t-1$) [16]. On peut dire qu'un circuit séquentiel = un circuit logique + une mémoire !!

le schéma d'un circuit séquentiel ressemble à la figure suivante :

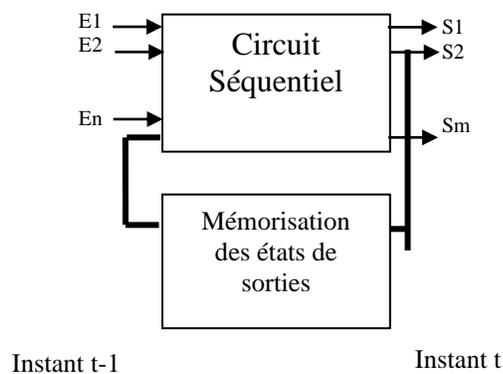


Figure 17: Schéma d'un circuit séquentiel.

Comme exemple de circuits séquentiels, on peut citer les bascules. Les bascules sont des circuits séquentiels qui possèdent deux états (Q et \bar{Q}).

2 Types de circuits séquentiels :

Il existe deux types de circuits séquentiels : synchrone et asynchrone [16-17]:

2.1 Circuits séquentiels asynchrone

Dans les circuits séquentiels asynchrones, le changement de la valeur de sortie est commandé par le changement des valeurs des entrées et celles des sorties actuelles. Car on a vu dans la définition des circuits séquentiels que les entrées seront combinées avec les états de sorties actuels pour calculer les valeurs des sorties à l'instant suivant.

Il existe plusieurs exemples de circuits séquentiels asynchrones, dans cette section nous allons étudier quelques uns :

2.1.1 La bascule RS :

Est une bascule qui contient deux entrées R et S. R pour dire Reset qui veut dire une mise à zero et S pour dire Set qui veut dire mise à un. La boîte noire de la bascule RS est donnée par le schéma suivant :

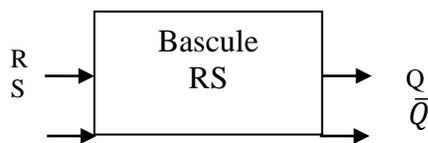


Figure 18: Boîte noire de la bascule RS asynchrone

Le tableau suivant résume le fonctionnement de la bascule RS

R	S	Q_n	\bar{Q}_n
0	0	Q_{n-1} (mémorisation de l'état précédente)	\bar{Q}_{n-1}
0	1	1 (mise à 1)	0
1	0	0 (mise à 0)	1
1	1	ϕ (cas interdit)	ϕ (cas interdit)

Tableau 6 : Fonctionnement de la bascule RS

La table de vérité de RS est la suivante :

R	S	Q_{n-1}	Q_n	\bar{Q}_n
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	ϕ	ϕ
1	1	1	ϕ	ϕ

Afin de construire le circuit logique de la bascule RS, nous tentons d'abord d'extraire la forme algébrique simplifiée de la sortie Q_n . la sortie Q_n est simplifiée par la méthode de karnaugh (voir les tableaux ci-dessous). Si on décide de dessiner le logigramme de RS avec uniquement les portes NOR, il est préférable d'extraire Q_n sous la forme POS (regroupement des zeros) (voir tableau 8), par contre la sortie \bar{Q}_n est extraire en faisant des

regroupement des uns des sorties Q_n puis on applique la négation à la fonction simplifiée (voir tableau 7).

Q_{n-1} \ RS	00	01	11	10
0	0	1	ϕ	0
1	1	1	ϕ	0

Tableau 7 : Fonction simplifiée sous forme SOP

Q_{n-1} \ RS	00	01	11	10
0	0	1	ϕ	0
1	1	1	ϕ	0

Tableau 8 : Fonction simplifiée sous forme POS

$$Q_n = S + \bar{R} \cdot Q_{n-1} \rightarrow$$

$$\begin{aligned} \overline{Q_n} &= \overline{S + \bar{R} \cdot Q_{n-1}} \\ &= \overline{S + \overline{\overline{\bar{R} \cdot Q_{n-1}}}} \\ &= \overline{S + \overline{R + \overline{Q_{n-1}}}} \end{aligned}$$

$$Q_n = \bar{R} \cdot (S + Q_{n-1})$$

$$\begin{aligned} &= \overline{\overline{\bar{R} \cdot (S + Q_{n-1})}} \\ &= \overline{R + \overline{S + Q_{n-1}}} \end{aligned}$$

Le circuit séquentiel de la bascule RS est donc donné par le schéma suivant:

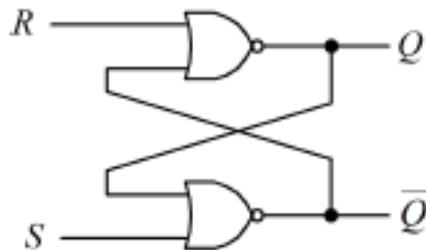


Figure 19: Logigramme de la bascule RS.

2.1.2 La bascule JK :

Cette bascule comporte deux entrées: J (Jack) et K (King). Son fonctionnement est décrit dans le tableau suivant :

J	K	Q_n	\bar{Q}_n
0	0	Q_{n-1}	\bar{Q}_{n-1}
0	1	0 (mise à 0)	1
1	0	1 (mise à 1)	0
1	1	\bar{Q}_{n-1} 1(basculement)	Q_{n-1}

Tableau 9 : Fonctionnement de la bascule JK

Nous remarquons que la bascule JK ressemble un peu à la bascule RS sauf pour le cas indéterminé de la bascule RS ($R=1$ et $S=1$) qui est pris en charge par la bascule JK en déclenchant un basculement de l'état de sortie (changement d'état). La table de vérité de RS est la suivante :

J	K	Q_{n-1}	Q_n	\bar{Q}_n
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

Réalisation de la bascule JK à partir des portes Nand.

Afin de faire un logigramme de la bascule JK, essayons d'abord de simplifier la sortie Q_n par le tableau de karnaugh à trois variables j, k et Q_{n-1} .

$Q_{n-1} \backslash JK$	00	01	11	10
0	0	0	1	1
1	1	0	0	1

$$\begin{aligned}
 Q_n &= J \cdot \bar{Q}_{n-1} + \bar{K} \cdot Q_{n-1} = J \cdot \bar{Q}_{n-1} + \bar{K} \cdot Q_{n-1} + \bar{Q}_{n-1} Q_{n-1} \\
 &= J \cdot \bar{Q}_{n-1} + (\bar{K} + \bar{Q}_{n-1}) \cdot Q_{n-1} \\
 &= J \cdot \bar{Q}_{n-1} + \overline{K \cdot Q_{n-1}} \cdot Q_{n-1} \\
 &= \overline{J \cdot Q_{n-1} + K \cdot Q_{n-1} \cdot Q_{n-1}} \\
 &= \overline{J \cdot Q_{n-1} + K \cdot Q_{n-1}}
 \end{aligned}$$

Pour avoir la sortie \bar{Q}_n , il suffit d'extraire la forme POS de Q_n puis appliquer la négation. Si on prends le même tableau de karnaugh précédent et on fait des regroupements de zeros, nous obtenons :

$$\begin{aligned}
 Q_n &= (J + \bar{Q}_{n-1}) \cdot (\bar{K} + Q_{n-1}) \Rightarrow \\
 \bar{Q}_n &= \overline{(J + \bar{Q}_{n-1}) \cdot (\bar{K} + Q_{n-1})} = \overline{(J + \bar{Q}_{n-1})} \cdot \overline{(\bar{K} + Q_{n-1})} \\
 \bar{Q}_n &= \overline{J} \cdot \overline{Q_{n-1} + 1} \cdot \overline{\bar{K} \cdot Q_{n-1}} = \overline{J} \cdot \overline{Q_{n-1} + 1} \cdot \overline{\bar{K} \cdot Q_{n-1}} \\
 \bar{Q}_n &= \overline{J + Q_{n-1}} \cdot \overline{Q_{n-1}} \cdot \overline{\bar{K} \cdot Q_{n-1}} = \overline{J + Q_{n-1}} \cdot \overline{Q_{n-1}} \cdot \overline{\bar{K} \cdot Q_{n-1}}
 \end{aligned}$$

Le circuit séquentiel de la bascule asynchrone JK réalisé à partir des portes Nand est donné par le schéma suivant :

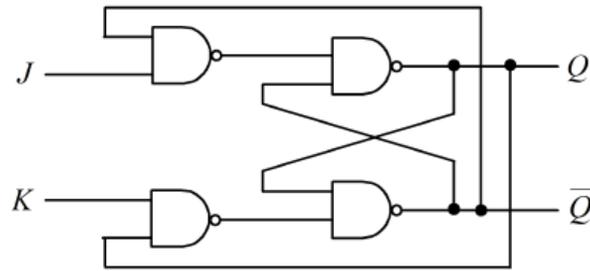


Figure 20: Logigramme de la bascule JK

2.1.3 La bascule D :

Dans la bascule D , on trouve une seule entrée, qui est notée D (pour Donnée ou Data). Son fonctionnement est le suivant : Si $D = 0$ alors la sortie $Q_n = 0$ sinon $Q_n = 1$. La sortie Q_n suit donc l'entrée D, par conséquent $Q_n = D$. sa table de vérité est donnée par :

D	Q_{n-1}	Q_n	\bar{Q}_n
0	0	0	1
0	1	0	1
1	0	1	0
1	1	1	0

Le logigramme de la bascule D :

Nous avons déjà vu que la sortie $Q_n = D$

Puisque la bascule D est un circuit séquentiel, nous allons donc introduire la sortie précédente dans la formule Q_n , nous aurons donc :

$$Q_n = D = D \cdot (Q_{n-1} + \bar{Q}_{n-1}) \text{ car } Q_{n-1} + \bar{Q}_{n-1} = 1$$

$$Q_n = D \cdot Q_{n-1} + D \cdot \bar{Q}_{n-1} \quad (1)$$

Reprenons maintenant la sortie de la bascule JK étudiée dans la section précédente :

$$Q_n = \bar{K} \cdot Q_{n-1} + J \cdot \bar{Q}_{n-1} \quad (2)$$

D'après les formules (1) et (2), si on pose $D=J$ et $\bar{D}=K$, nous remarquons que la bascule D est une bascule JK à laquelle on a ajouté un inverseur entre les entrées J et K. La bascule D est représentée par le schéma suivant.

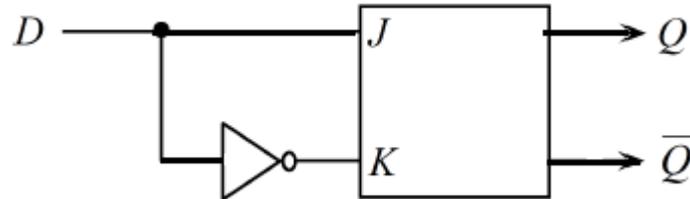


Figure 21: Réalisation de la bascule D en fonction de la bascule JK.

2.2 Circuits séquentiels synchrones

Les bascules synchrones sont des bascules dont le changement d'état de sortie (Q_n) ne peut s'effectuer que lorsqu'un signal de contrôle appelé signal d'horloge est déclenché [16].

2.2.1 Horloge :

une horloge, notée H ou CK(clock en anglais) est une variable logique qui passe de zéro à un et de un à zéro de manière périodique. Elle est utilisée comme entrée dans les circuits séquentiels synchrones.

L'horloge est représentée soit par un signal impulsionnel ou périodique. La figure suivante montre le signal périodique d'une horloge H.

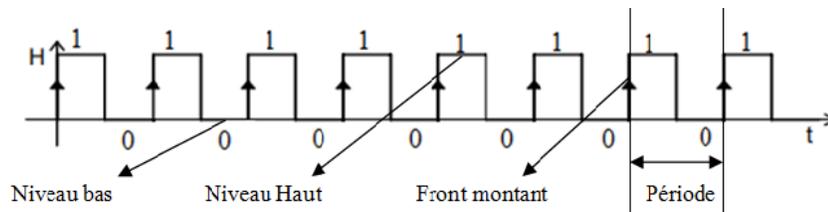


Figure 22: Signal périodique d'une horloge.

La durée entre deux fronts successifs (ou deux niveaux) s'appelle période (T) qui est calculée en seconde.

Fréquence = $1/\text{période}$ = nombre de changement par seconde, elle est calculée en hertz (Hz). Une horloge de 1 hertz a une période de 1 seconde tandis qu'une horloge de 1 megahertz a une période de 1 microseconde.

2.2.2 Type de synchronisation :

Il existe deux type de bascules synchrones : les bascules synchronisés sur niveau et les bascules synchronisés sur front.

Les bascules synchronisées sur niveau :

on distingue deux types de synchronisation : sur niveau haut et sur niveau bas .

➤ synchronisation sur niveau haut

Dans ce cas, la bascule ne fonctionne normalement que lorsque $H=1$ (Niveau haut). Dans ce cas un changement dans les valeurs d'entrées entraîne un changement dans la valeur de sortie. Si $H=0$ (Niveau bas) alors la sortie Q garde l'état précédent.

➤ synchronisation sur niveau bas

Dans ce cas, la bascule ne fonctionne normalement que lorsque $H=0$ (Niveau bas). Dans ce cas un changement dans les valeurs d'entrées entraîne un changement dans la valeur de sortie. Si $H=1$ alors la sortie Q garde l'état précédent.

Inconvénients des bascules synchronisées sur niveau :

Dans ce type de synchronisation, la valeur de sortie est modifiée à chaque changement des valeurs d'entrées pendant toute la durée de l'état de l'horloge pour niveau haut (ou 0 pour le niveau bas). Si, pendant cette durée, des parasites apparaissent sur les entrées, ceux-ci peuvent entrainer des changements d'état imprévus sur la sortie Q. Afin de minimiser au maximum la durée de cet état sensible, on s'arrange pour que la bascule reste dans son état

mémoire sauf pendant un bref instant, juste au moment où l'entrée passe de 0 à 1 (ou de 1 à 0). La bascule est dite synchronisée sur front

Les bascules synchronisés sur front :

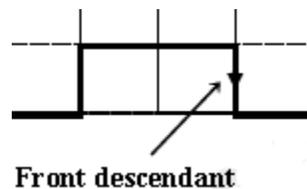
On distingue deux types de synchronisation sur front: sur front montant et sur front descendant

➤ les bascules actives sur front montant :

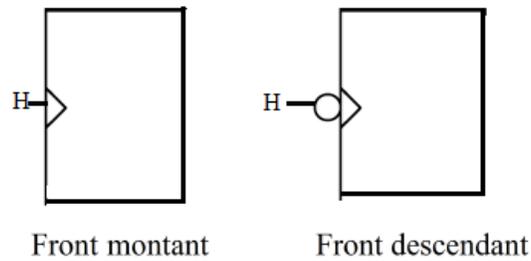
Le passage du niveau bas (état 0) au niveau haut (état 1) est appelé : front montant. Dans ce type de bascule, la sortie Q fonctionne normalement à chaque front montant de H. Si H n'est pas sur un front montant, la sortie Q ne change pas (mémorisation de l'état précédent).

➤ les bascules actives sur front descendant :

Le passage du niveau haut (état 1) au niveau bas (état 0) est appelé : front descendant. Dans ce type de bascule, la sortie Q fonctionne normalement à chaque front descendant de H. Si H n'est pas sur un front descendant, la sortie Q ne change pas (mémorisation de l'état précédent).



L'horloge H est utilisée comme entrée dans les circuits séquentiels synchrones, sa représentation diffère selon le type de synchronisation utilisé (voir schéma ci-dessous).



2.2.3 La bascule RS synchrone (RSH)

Cas front montant : si $H=0$ alors $Q_n = Q_{n-1}$ (Mémorisation)

Sinon fonctionnement normale de la bascule RS.

La boîte noire ainsi que le logigramme de la bascule RSH sont présentés par les schémas suivants

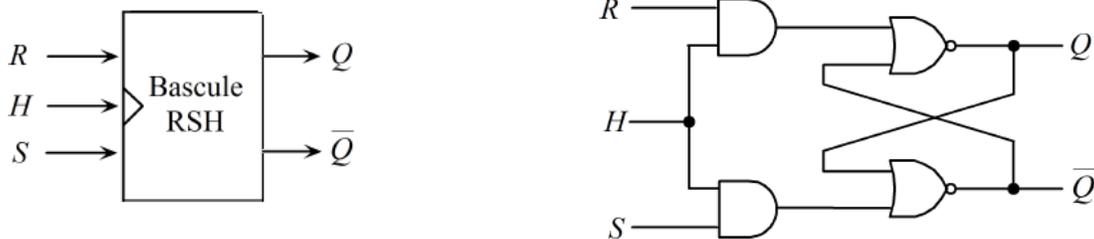


Figure 23: Logigramme de la bascule RS en front montant

Cas front descendant : si $H=1$ alors $Q_n = Q_{n-1}$ (Mémorisation)

Sinon fonctionnement normale de la bascule RS.

La boîte noire ainsi que le logigramme de la bascule RSH en mode descendant sont présentés par les schémas suivants

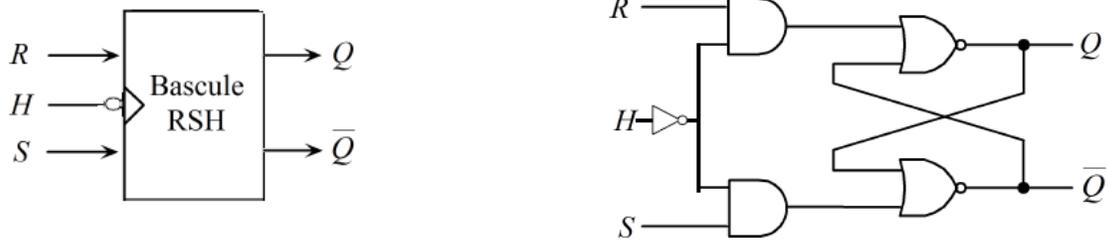


Figure 24: Logigramme de la bascule RS en mode descendant

2.2.4 La bascule JK synchrone (JKH)

Cas de synchronisation sur niveau haut: Si $H=0$ alors $Q_n = Q_{n-1}$ (Mémorisation)

Sinon fonctionnement normalement de la bascule JK.

le logigramme de la bascule JKH en mode montant est présenté par le schéma suivant :

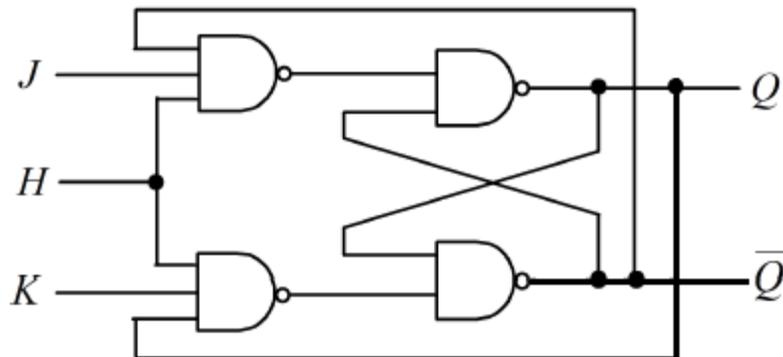


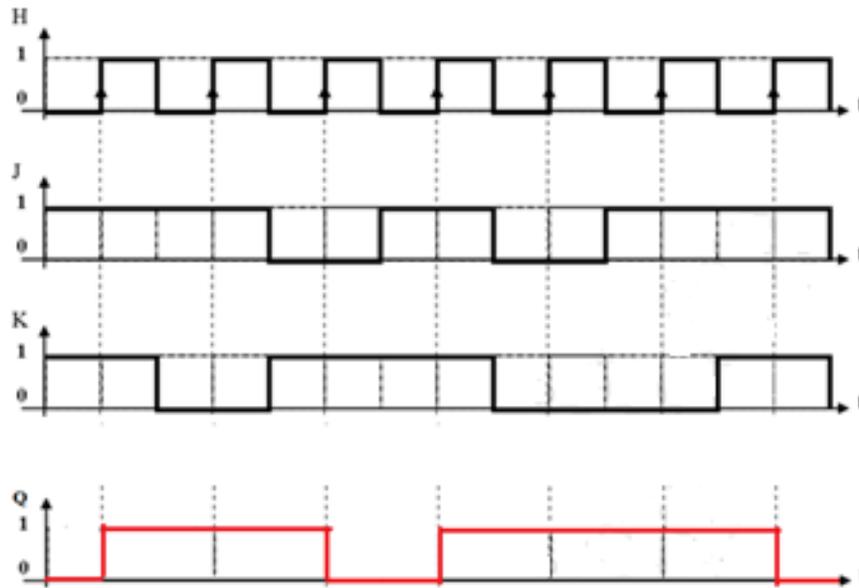
Figure 25: Logigramme de la bascule JK synchrone.

L'inconvénient majeur de ce type de synchronisation est que pendant toute la durée du niveau haut, l'état de sortie Q_n bascule entre 0 et 1 d'où l'utilisation d'une bascule JK synchronisé sur front s'avère plus avantageux.

La table suivante montre le fonctionnement de la bascule JK sur front montant

Les entrées				Sortie	Etat
H	J	K	Q_{n-1}	Q_n	
0	∇	∇	Q_{n-1}	Q_{n-1}	Mémorisation
1 (Niveau haut)	∇	∇	Q_{n-1}	Q_{n-1}	Mémorisation
\downarrow (front descendant)	∇	∇	Q_{n-1}	Q_{n-1}	Mémorisation
\uparrow (front montant)	0	0	Q_{n-1}	Q_{n-1}	Mémorisation
\uparrow (front montant)	0	1	Q_{n-1}	0	Mise à zero
\uparrow (front montant)	1	0	Q_{n-1}	1	Mise à un
\uparrow (front montant)	1	1	Q_{n-1}	Q_{n-1}	\bar{Q}_{n-1}

Le chronogramme suivant montre le fonctionnement de la bascule JK après chaque front montant :



Nous remarquons que dans le premier front montant, $J=1$ et $K=1$ ce qui déclenche un basculement de la sortie Q ($Q=\bar{0}=1$). La sortie reste à l'état 1 jusqu'au prochain front montant.

Dans le deuxième front montant, $J=1$ et $K=0$, c'est-à-dire la sortie Q est mise à un. Puisque Q avait déjà la valeur un donc elle va garder cette valeur jusqu'au prochain front montant et ainsi de suite.

Ce qu'il faut retenir dans la synchronisation front (montant ou descendant) est que l'état reste stable entre les deux fronts successifs. La durée entre deux fronts successifs s'appelle période (T).

2.2.5 La bascule D synchrone:

Le fonctionnement de la bascule D synchrone sur niveau haut est le suivant :

Si $H=0$ alors $Q_n=Q_{n-1}$ (état de mémorisation)

Sinon $Q_n=D$

La table de vérité de la bascule D synchronisée sur niveau haut est la suivante :

Entrées		Sortie
H	D	Q _n
0	∇	Q _{n-1} (Etat précédent)
1	0	0
1	1	1

La table peut être encore réduite, ce qui donne

H	Q _n
0	Q _{n-1}
1	D

$$Q_n = \bar{H} \cdot Q_{n-1} + H \cdot D$$

Nous avons déjà démontré dans la section précédente que la bascule D peut être représenté par des entrées JK. Même principe est utilisé pour la bascule D synchrone. Le logigramme de la bascule D en utilisant les portes Nand est représenté par les entrées J et K comme suit :

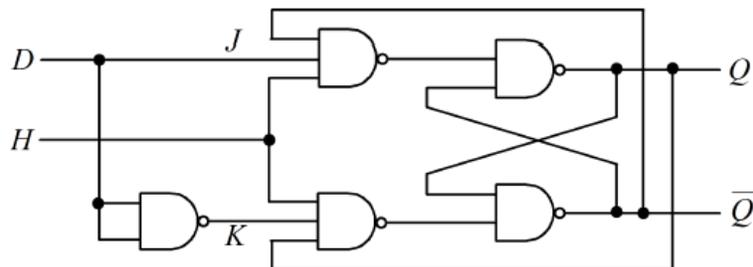


Figure 26: Logigramme de la bascule D synchrone.

3 Les registres

3.1 Définition

Un registre à n bit est un circuit séquentiel synchrone composé de n bascules synchrones reliées l'une à la suite de l'autre formant ainsi un mot de n bits. Ces bascules sont commandées par le même signal d'horloge. La taille du registre correspond donc au nombre de bascules que contient le registre. Un registre permet de mémoriser (sauvegarder) une information sur n bits [16-18].

3.2 Fonctionnement des registres

Dans cette section, on va étudier les deux modes de fonctionnement des registres : les registres à chargement série ou registres à décalage et les registres à chargement parallèle, appelés aussi registres de mémorisation.

3.2.1 *Registre à décalage (Registre série):*

Le registre à décalage, comme son nom l'indique, consiste à faire un décalage ou un glissement de l'information bit par bit vers la gauche ou vers la droite. Une application importante des registres à décalage est la transmission série de données logiques.

Décalage à droite :

Dans ce type de registre, la sortie de la bascule synchronisée i est reliée à l'entrée de la bascule synchronisée $i+1$. L'information est donc décalée vers la droite. L'état de la première bascule se décale aux bascules suivantes d'où le nom de «circuits à décalage». Voici un exemple d'un registre à décalage à droite à 4 bits (on a pris l'exemple de 4 bascules D synchronisées sur front montant) :

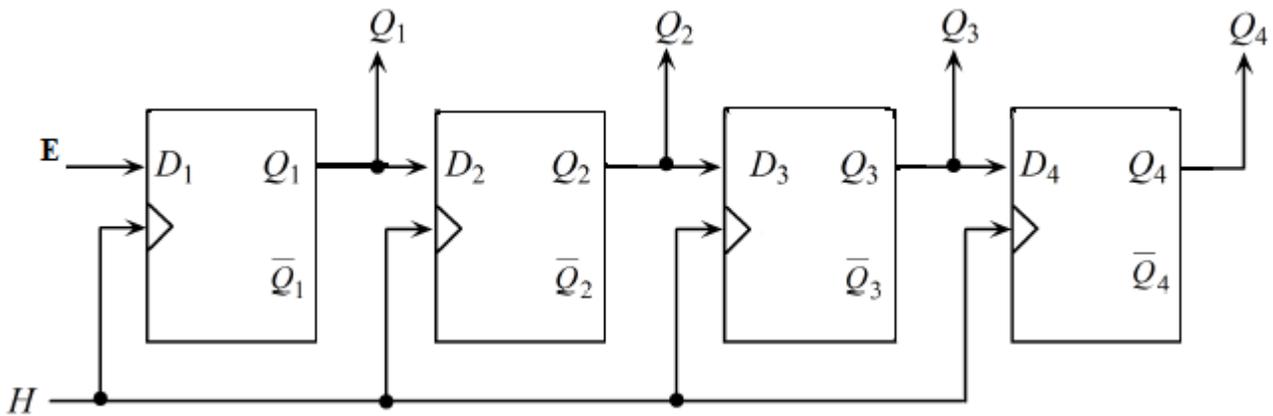


Figure 27: Registre à décalage à droite à 4 bits.

Registre à décalage à gauche :

Dans ce type de registre, la sortie de la bascule $i+1$ est reliée (donc recopié) à l'entrée de la bascule i . En appliquant le même signal d'horloge à toutes les bascules que contient le registre, l'état de la dernière bascule se décale aux bascules précédentes d'où le nom de «circuits à décalage à gauche». Voici un exemple d'un registre à décalage à gauche à 4 bascules D synchronisées sur front montant.

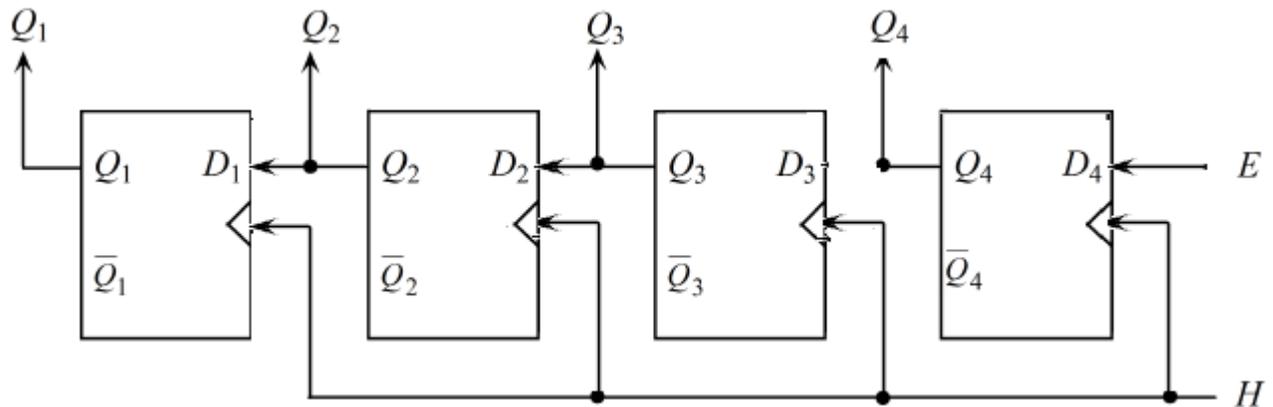


Figure 28: Registre à décalage à gauche à 4 bits (04 bascules).

Remarque : il existe aussi des registres à décalage mixte dites réversible où le décalage s'effectue vers la droite ou vers la gauche.

3.2.2 *Registre à chargement parallèle :*

Un registre à chargement parallèle ou registre de mémorisation ou de données est un registre qui contient un ensemble de bascules synchrones reliées en parallèle. Donc en plaçant plusieurs registres à 1 bit (bascule) en parallèle, On obtient ainsi un registre à chargement parallèle. Ce type de registre permet de charger une information sur n bits en même temps (en parallèle), ainsi les bits (ou les différents étages) sont indépendants les uns des autres (certains signaux agissent sur l'ensemble des étages, tel que remise à 0 et remise à 1). Ce registre possède une entrée de chargement chg. Son fonctionnement est le suivant :

Si $chg=0$ alors mémorisation des états précédents.

Sinon un chargement

La figure suivante illustre le principe de fonctionnement d'un registre à chargement parallèle. La figure montre un registre à 4 bits, actif sur niveau, à chargement parallèle.

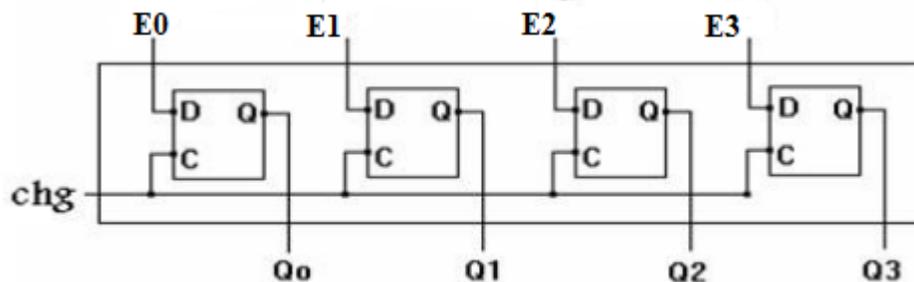


Figure 29: Registre à chargement parallèle à 4 bits.

les 4 bascules sont chargées en parallèle et lues en parallèle en synchrone avec le signal d'horloge.

Les registres parallèles sont très utilisés comme circuit de temporisation ou comme circuit de mémoire.

4 Analyse d'un circuit séquentiel (automates):

4.1 Définition :

Un automate d'état fini possède un nombre fini d'éléments et de mémoires (bascules). Il prends 2^n états appelés internes où n est le nombre de bits de mémoires (c'est-à-dire le nombre de bascules) [18].

Un automate est caractérisé par :sa matrice de transition et son graphe d'état.

L'analyse d'un circuit séquentiel consiste à trouver les équations caractéristiques décrivant son fonctionnement. Il s'agit d'exprimer les états futures ($Q+$) et les sorties du circuits (S) en fonction des entrées (E) et les états présents (Q). l'analyse passe par 4 étapes :

- Déterminer les equations des entrees des bascules,
- Déterminer les equations de sorties des bascules,
- Déterminer les equations de sorties en fonction des entrees et des etats actuels des bascules,
- Tracage du diagramme de transition.

4.2 Table de transition

La table de transition d'une bascule définit les valeurs des entrées synchrones qui ont déclenchées les états de sortie. Il s'agit de déterminer les valeurs des entrées sachant l'état actuel et l'état précédent.

Ci-dessous la table de transition des trois bascules RS, JK et D.

Q_{n-1}	Q_n	R	S	J	K	D
0	0	X	0	0	X	0
0	1	0	1	1	X	1
1	0	1	0	X	1	0
1	1	0	X	X	0	1

Par exemple pour $Q_{n-1}=0$ et $Q_n=0$, deux cas sont possibles pour la bascule RS :

- Les entrées RS peuvent avoir les valeurs 10 (R=1 et S=0) car la sortie $Q_n=0$ (Remise à zéro).
- Les entrées RS peuvent avoir les valeurs 00 (R=0 et S=0) car la sortie $Q_n=Q_{n-1}$ (Mémorisation).

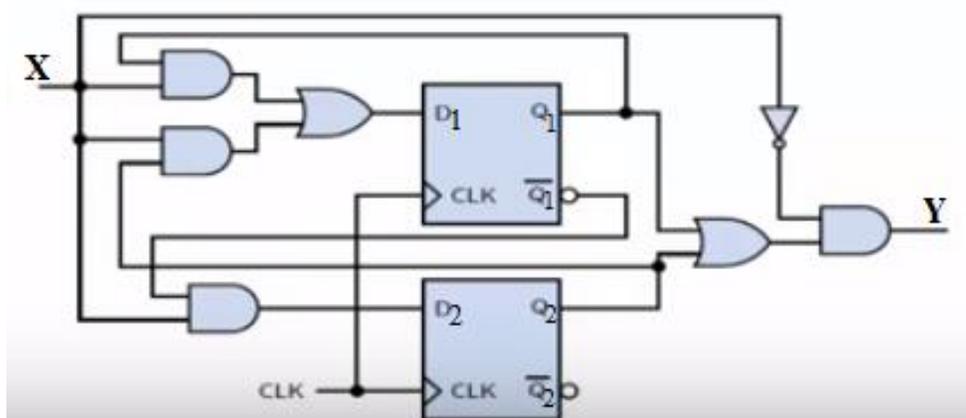
Prenons un deuxième exemple pour $Q_{n-1}=1$ et $Q_n=0$, deux cas sont possibles pour la bascule JK :

- Les entrées JK peuvent avoir les valeurs 01 (J=0 et K=1) car la sortie $Q_n=0$ (Remise à zéro).
- Les entrées JK peuvent avoir les valeurs 11 (J=1 et K=1) car la sortie $Q_n=\overline{Q_{n-1}}$ (Mémorisation).

4.3 Automate de Mealy

Si les valeurs des variables de sortie dépendent de l'état présent et des variables d'entrée, le système séquentiel est appelé une machine de Mealy. Dans ce cas, le changement des sorties n'est pas synchrone: il se fait avec le changement des entrées.

Exemple : analyser le circuit suivant :



Etape1 :

Exprimons d'abord les entrées des bascules (D1 et D2) en fonction de l'entrée X et les états précédents (Q1 et Q2) :

$$D1 = X.Q1 + X.Q2 = X.(Q1 + Q2)$$

$$D2 = X.\overline{Q_1}$$

Etape2 :

Exprimons les sorties des bascules en fonction de l'entrée X et les états précédents (Q1 et Q2). Notons pour la bascule D, l'état futur (Q^+) est égal à l'entrée ($Q^+ = D$). Donc pour chaque bascule D, nous avons : $Q_1^+ = D_1$ et $Q_2^+ = D_2$

Ce qui donne :

$$Q_1^+ = D_1 = X.(Q1 + Q2)$$

$$Q_2^+ = D_2 = X.\overline{Q_1}$$

Etape3 :

déterminons maintenant la sortie du circuit (Y) :

$$Y = \overline{X}.(Q1 + Q2)$$

Etape4 :

Traçons maintenant la table de vérité (table d'état) de ce circuit à trois entrées (X, Q1, Q2):

Entrée	Etats précédents		Etats futurs		Sortie
	Q1	Q2	Q1 ⁺	Q2 ⁺	
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	0	1
0	1	1	0	0	1
1	0	0	0	1	0
1	0	1	1	1	0
1	1	0	1	0	0
1	1	1	1	0	0

Etape5 :

Traçons maintenant le diagramme ou le graphe de transition.

Dans ce circuit nous avons deux bascules donc il y a $2^2=4$ états possibles :

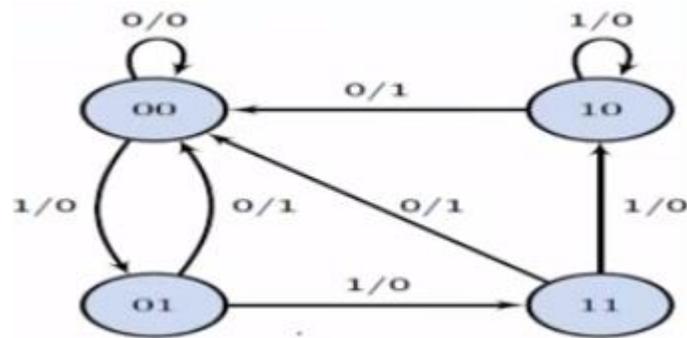
Etat0 =00, Etat 1=01, Etat2=10 et Etat3=11

Chaque état est représentée par un cercle dans le graphe d'état.dans cet exemple nous dessinons donc 4 cercles (voir figure ci_dessous).

Chaque ligne dans la table d'état (voir etape 4) représente une flèche dans le diagramme de transition.

Sur la flèche j'écris les valeurs correspondantes à X/Y (entrée/sortie). Par exemple pour la première ligne de la table d'état, le passage de l'état $(Q_1Q_2)=00$ vers l'état suivant $(Q_1^+ Q_2^+)=00$ est repretée par une flèche et j'écris sur la flèche (0/0). (l'écriture 0/0 représente les valeurs de X/Y)

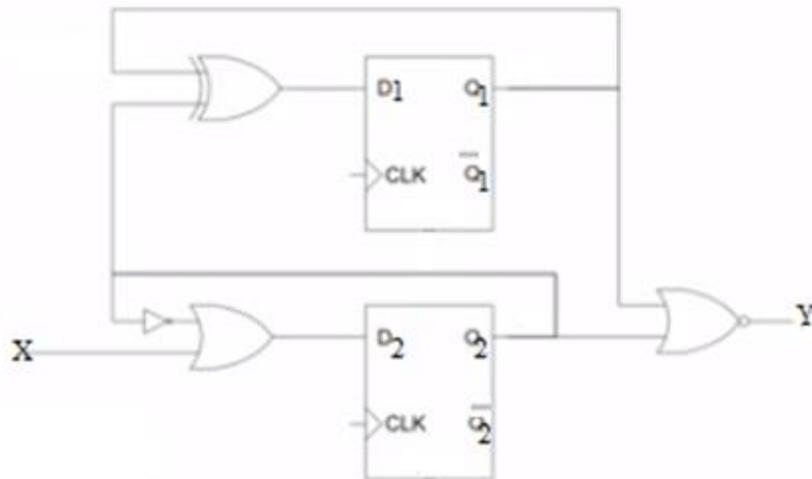
On obtient donc le graphe de transition suivant :



4.4 Automate de Moore

Si les valeurs des variables de sortie dépendent uniquement de l'état présent, le système séquentiel est appelé une machine de Moore. Dans ce cas, le changement des sorties est synchrone: il se fait avec le changement des états.

Exemple : analyser le circuit suivant :



Étape1 :

Exprimons d'abord les entrées des bascules (D1 et D2) en fonction de l'entrée X et les états précédents (Q1 et Q2) :

$$D1 = Q1 \oplus Q2$$

$$D2 = X + \overline{Q2}$$

Etape2 :

Exprimons les sorties des bascules en fonction de l'entrée X et les les états précédents (Q1 et Q2) :

Pour chaque bascule D, nous avons : $Q_1^+ = D_1$ et $Q_2^+ = D_2$

Ce qui donne :

$$Q_1^+ = D_1 = Q_1 \oplus Q_2$$

$$Q_2^+ = D_2 = X + \overline{Q_2}$$

Etape3 :

déterminons maintenant la sortie du circuit (Y) :

$$Y = \overline{Q_1} + Q_2$$

Nous remarquons que la sortie Y dépend uniquement des états précédents et ne dépend de l'entrée X

Etape4 :

L'étape 4 consiste à dresser la table d'état, pour cela , on doit énumérer tous les cas possibles des états précédents (Q1,Q2) et de l'entrée X, nous obtenons le tableau suivant :

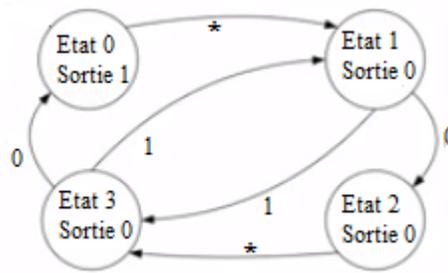
N°	Entrée	Etats précédents		Etats futurs		Sortie
Ligne	X	Q1	Q2	Q1 ⁺	Q2 ⁺	Y
1	0	0	0	0	1	1
2	0	0	1	1	0	0
3	0	1	0	1	1	0
4	0	1	1	0	0	0
5	1	0	0	0	1	1
6	1	0	1	1	1	0
7	1	1	0	1	1	0
8	1	1	1	0	1	0

Etape5 :

Traçons maintenant le diagramme ou le graphe de transition. Dans ce circuit nous avons deux bascules donc il y a $2^2=4$ états possibles :

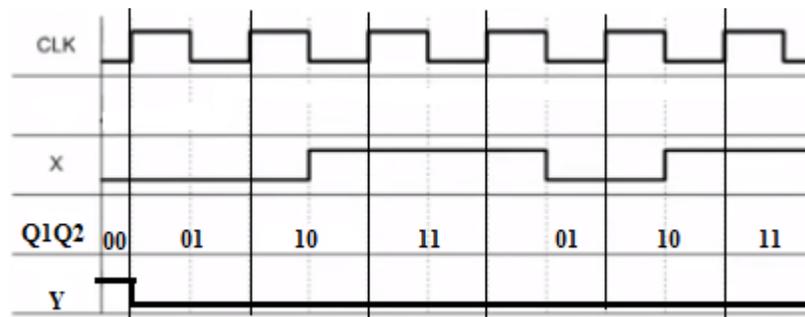
Etat1 =00, Etat 2=01, Etat3=10 et Etat4=11,

Ces états seront représentés par des cercles. Puisque la sortie du circuit dépend uniquement des états , donc on écrit dans le cercle la sortie correspondante (voir le diagramme d'état ci-dessous). Rappelons que chaque ligne dans la table d'état représente une flèche dans le diagramme de transition. Sur la flèche j'écris les valeurs de l'entrée X correspondantes. Le symbole * signifie que l'entrée X peut prendre soit la valeur 1 ou 0. Par exemple le passage de l'état 0 ($Q_1Q_2=00$) à l'état 1 ($Q_1^+ Q_2^+ =01$) correspond aux lignes 1 et 5 de la table d'état. Dans ce cas, pour la ligne 1 la valeur de X est égale à zéro tandis que pour la ligne 5, elle est égale à 1.



A partir du diagramme d'état, il est facile de représenter le chronogramme d'un circuit séquentiel

Ci-dessous le chronogramme du circuit étudié sur front montant.



Remarque :

Tout problème peut être résolu par une machine de type Mealy ou Moore.

Pour un même problème, une machine de Mealy demande moins d'états que la machine de Moore.

Rappel

Quel que soit le type de machine (Moore ou Mealy) l'état suivant du système $Q_i (n+1)$ dépend de l'état actuel $Q_i (n)$ et des entrées $E_i (n)$. La différence entre machine de Mealy et machine de Moore n'intervient que sur les sorties. Dans une machine de Mealy, les sorties $S_i (n)$ dépendent de l'état $Q_i (n)$ et des entrées $E_i (n)$. Par contre dans une machine de Moore, les sorties $S_i (n)$ ne dépendent que de l'état $Q_i (n)$.

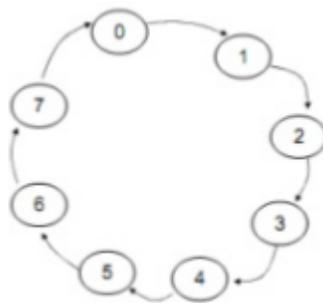
5 Les compteurs / décompteurs:

Un compteur (ou décompteur) est un circuit électronique constitué essentiellement par un ensemble de bascules. Il permet de comptabiliser le nombre d'événements qui se produisent pendant un temps donné. L'information disponible est située sur l'ensemble des sorties des bascules.

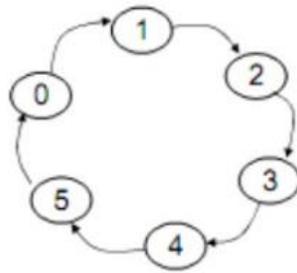
Un compteur (décompteur) modulo N est un circuit séquentiel qui permet de compter de 0 à $N-1$ (de $N-1$ à 0) tel que N est un entier positif et $2^{n-1} < N \leq 2^n$. Dans ce cas le compteur (décompteur) nécessite n bascules pour faire le comptage qui seront reliées entre elles par un circuit logique.

Si N s'écrit sous forme de 2^n , on parle d'un compteur (décompteur) à cycle complet (par exemple un compteur (décompteur) modulo 4, 8 ou 16) sinon, il s'agira d'un compteur (décompteur) à cycle incomplet (par exemple un compteur (décompteur) modulo 5,7,9..).

Le cycle complet suivant représente un compteur modulo 8 qui permet de compter de $000_{(2)}$ jusqu'à $(111)_{(2)}$ c'est-à-dire de 0 à 7 puis revient à 0.



Le compteur modulo 6 suit un cycle incomplet car il permet de compter de (000) jusqu'à (101) puis revient à (000) et donc les combinaisons 110 et 111 ne sont pas prises en compte par ce compteur.



On distingue deux types de compteurs (décompteurs) : compteurs (décompteurs) synchrones et asynchrones. Dans les compteurs (décompteurs) synchrones toutes les bascules seront reliées au même signal d'horloge tandis que dans les compteurs (décompteurs) asynchrones seulement la première bascule est reliée à l'horloge, toutes les bascules qui suivent celle-ci sont commandées par la bascule précédente.

Exemple 1 : réalisation d'un compteur synchrone modulo 4 par les bascules D

Un compteur synchrone modulo $4=2^2$ nécessite donc 2 bascules D, il s'agit d'un compteur à cycle complet qui permet de compter de 0 jusqu'à 3 (4-1).

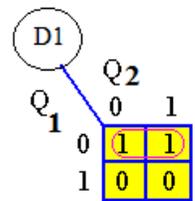
Notons que la table de transition de la bascule D est donnée par :

Q	Q ⁺	D
0	0	0
0	1	1
1	0	0
1	1	1

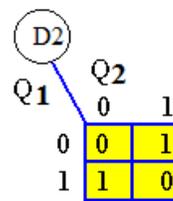
Soit Q_i =état précédent de la bascule D_i et Q_i^+ = état actuel de la bascule D_i . La table de transition des deux bascules D_1 et D_2 prises en même temps est la suivante :

Q_2	Q_1	Q_2^+	Q_1^+	D2	D1
0	0	0	1	0	1
0	1	1	0	1	0
1	0	1	1	1	1
1	1	0	0	0	0

Les fonctions logiques de D1 et D2 en fonction des états Q_2 et Q_1 sont calculées par les 2 tableaux de karnaugh suivants :



$$D_1 = \overline{Q_1}$$



$$D_2 = \overline{Q_1}Q_2 + Q_1\overline{Q_2}$$

$$D_2 = Q_1 \oplus Q_2$$

Enfin le compteur synchrone modulo 4 est représenté par le circuit séquentiel suivant :

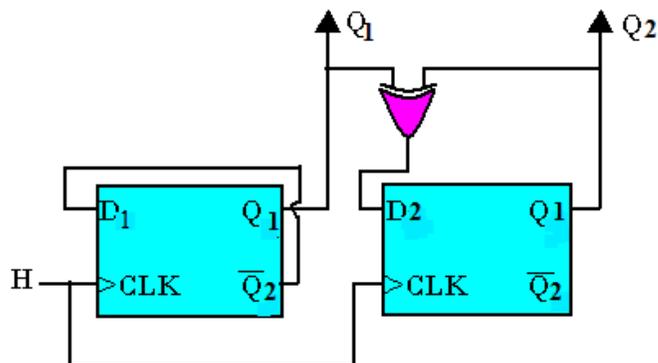


Figure 30: schéma d'un compteur synchrone modulo 4 par les bascules D

Exemple 2 : réalisation d'un décompteur synchrone modulo 8 par les bascules D

L'étude d'un décompteur se fait en suivant le même principe que celui d'un compteur. Un décompteur synchrone modulo $8=2^3$ nécessite donc 3 bascules D, il s'agit d'un compteur à cycle complet qui permet de compter de 0 jusqu'à 7.

La table de transition des trois bascules D_3 , D_2 et D_1 est la suivante :

Q_3	Q_2	Q_1	Q_3^+	Q_2^+	Q_1^+	D_3	D_2	D_1
1	1	1	1	1	0	1	1	0
1	1	0	1	0	1	1	0	1
1	0	1	1	0	0	1	0	0
1	0	0	0	1	1	0	1	1
0	1	1	0	1	0	0	1	0
0	1	0	0	0	1	0	0	1
0	0	1	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1

Les fonctions logiques de D_0 , D_1 et D_2 en fonction des états Q_2 , Q_1 et Q_0 sont calculées par les 3 tableaux de karnaugh suivants :

Q_3Q_2 Q_1	00	01	11	10
0	1	1	1	1
1	0	0	0	0

$$D_1 = \overline{Q_1}$$

Q_3Q_2 Q_1	00	01	11	10
0	1	0	0	1
1	0	1	1	0

$$D_2 = \overline{Q_1} \cdot \overline{Q_2} + Q_2 \cdot Q_1 = \overline{Q_1} \oplus Q_2$$

Q3Q2 Q1	00	01	11	10
0	1	0	1	0
1	0	0	1	1

$$\begin{aligned}
 D3 &= Q1.Q3 + Q3.Q2 + \overline{Q1}.\overline{Q2}.\overline{Q3} \\
 &= (Q1+Q2) . Q3 + \overline{(Q1 + Q2)} . \overline{Q3} \\
 &= \overline{(Q1 + Q2)} \oplus Q3
 \end{aligned}$$

Le décompteur synchrone modulo 8 est représenté par le circuit séquentiel suivant :

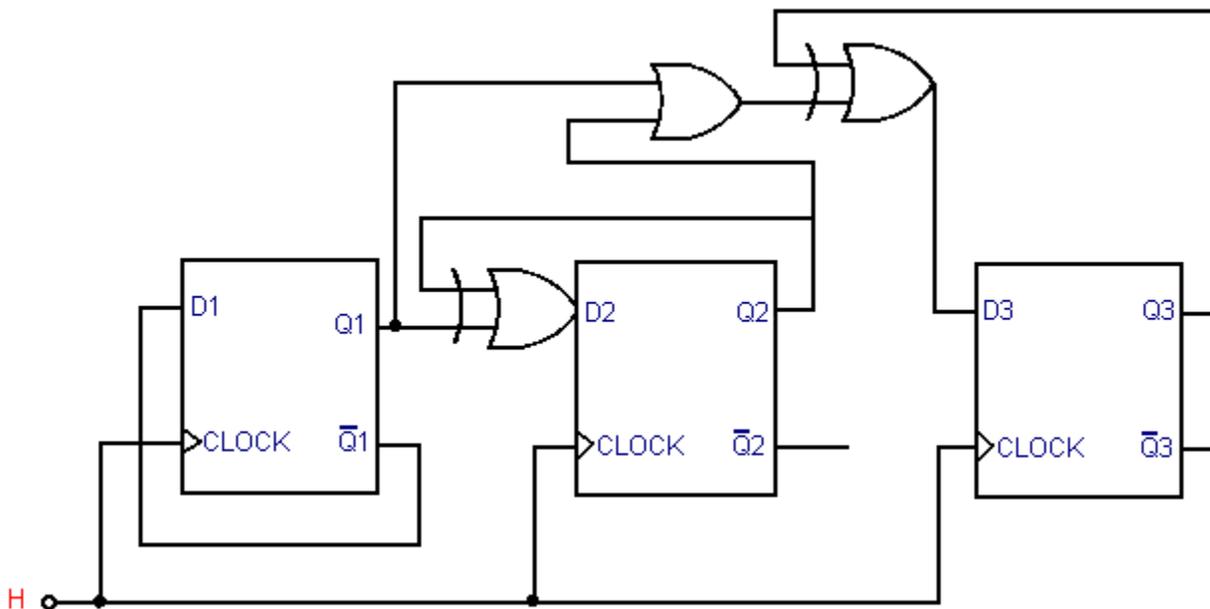


Figure 31: schéma d'un décompteur synchrone modulo 8 par les bascules D

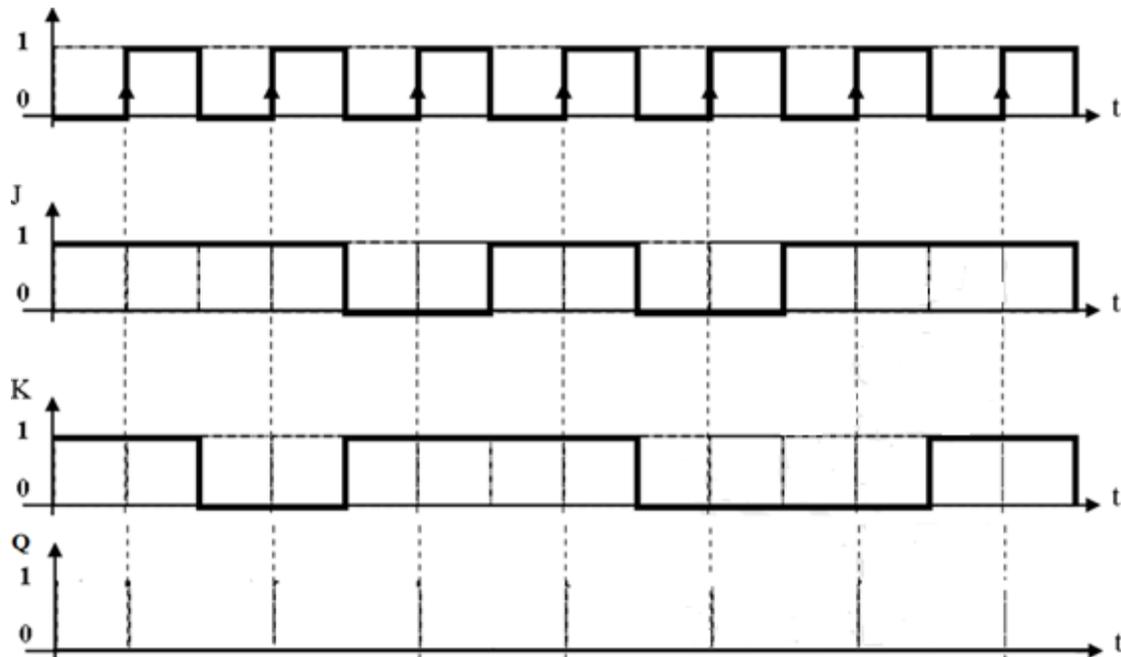
6 Conclusion :

Dans ce chapitre, nous avons parlé des circuits séquentiels synchrones et asynchrones. Nous avons vu aussi les différentes démarches adoptées afin d'analyser un circuit séquentiel. Enfin nous avons donné quelques exemples de ces circuits à savoir les registres et les compteurs.

7 Exercices

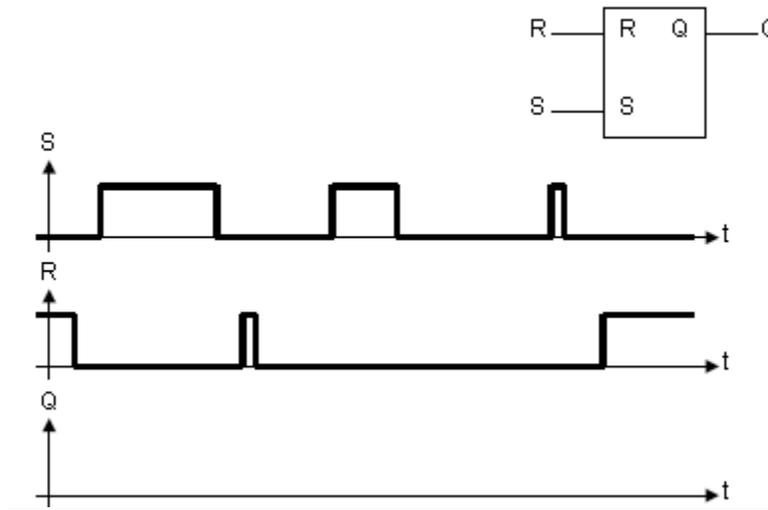
Exercice 1:

- 1/ Donner la TV de la bascule JK asynchrone.
- 2/ Compléter le chronogramme de la bascule JK asynchrone en fonction des entrées sachant que initialement la sortie Q est mise à zéro ($Q=0$).
- 3/ Donner la TV de la bascule JK synchrone (JKH).
- 4/ Compléter le chronogramme de la bascule JKH sur le niveau haut.
- 5/ La durée entre deux niveaux successifs s'appelle période (T), donner la fréquence de cette horloge en Hz puis en Mhz sachant que la période =2ms.

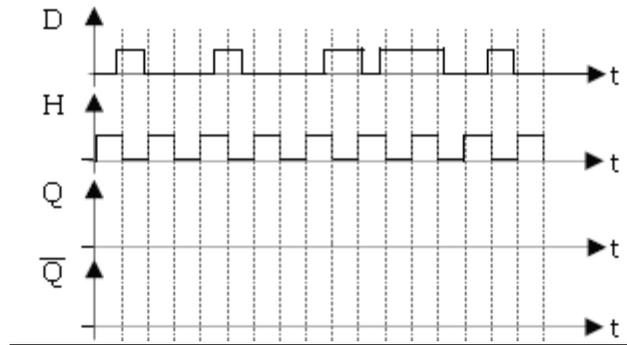


Exercice 3 :

Compléter le chronogramme ci-dessous

**Exercice 4:**

Compléter les chronogrammes suivants (appliqués à une bascule D active sur front montant) :

**Exercice 5:**

Rappel :

La table de transition d'une bascule définit les valeurs des entrées synchrones qui ont provoqués le basculement. Il s'agit de déterminer les valeurs des entrées (RS, JK et D) sachant qu'on connaît déjà les valeurs des sorties.

1. Donner la Table de transition d'une bascule RS
2. Donner la Table de transition d'une bascule JK
3. Donner la Table de transition d'une bascule D

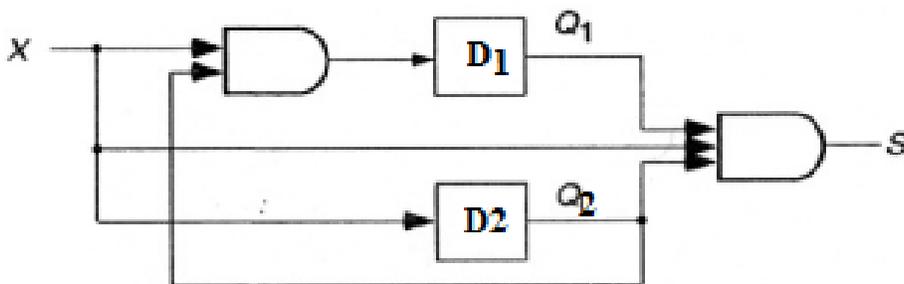
Exercice 6 :

Faites la réalisation d'un compteur synchrone modulo 5 (cycle incomplet), en utilisant des bascules D à front montant.

Exercice7 :

Analyser le circuit suivant :

1. déterminer la table des états
2. Donner son diagramme de transition.



Corrigé des Exercices

1 Chapitre 1

Exercice 1

$$F0 = x(\bar{x} + y) = x + y$$

$$F1 = x + \bar{x}.y = x + y$$

$$F2 = x.y + \bar{x} = \bar{x} + y$$

$$F3 = (x + y). (x + z). (y + z) = x.y + x.z + y.z$$

$$F4 = \overline{\bar{x}.y} + \bar{y} = \bar{x} + \bar{y}$$

$$F5 = \overline{\overline{\bar{x} + y}} + x.y. (\bar{z} + \bar{y}) = x + y$$

$$F6 = \overline{\bar{x} + y + z}. t + \overline{\bar{x} + y}. z = \bar{x}. \bar{y}.z + \bar{x}. \bar{y}.t$$

$$F7 = x.(x. \bar{y}(y + x.t)) + \bar{x} + \bar{t} = \bar{x} + \bar{y} + \bar{t}$$

$$F8 = \overline{\overline{\bar{x} + y}} + z + \overline{\overline{\bar{x} + y}}.z = x + y + \bar{z}$$

$$F9 = (\bar{x} + \bar{y}). (x + y) = x \oplus y$$

$$F10 = (\bar{x} + y). (x + \bar{y}) = \overline{\bar{x} \oplus \bar{y}}$$

Exercice 2 :

1) L'expression de $S1 = \bar{x}.y + z$

TV (S1)

X	Y	Z	\bar{x}	$\bar{x}.y$	S1
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	0	0	0
1	1	1	0	0	1

S1 simplifiée = $S1 = \bar{x}.y + z$

2) L'expression de $S2 = \overline{\bar{x} + \bar{y}} + \overline{\bar{x}. \bar{y}}$

TV (S2)

X	y	\bar{x}	\bar{y}	$\bar{x} + \bar{y}$	$\bar{x}. \bar{y}$	$\overline{\bar{x} + \bar{y}}$	$\overline{\bar{x}. \bar{y}}$	S2
0	0	1	1	1	1	0	0	0
0	1	1	0	1	0	0	1	1
1	0	0	1	1	0	0	1	1
1	1	0	0	0	0	1	1	1

Après l'application de la simplification algébrique : $S2 = x + y$

3) L'expression de $S3 = \overline{x + (y \oplus z)} . x$

TV (S3)

x	y	Z	$y \oplus z$	$x + (y \oplus z)$	$\overline{x + (y \oplus z)}$	S3
0	0	0	0	0	1	0
0	0	1	1	1	0	0
0	1	0	0	0	1	0
0	1	1	1	1	0	0
1	0	0	1	1	0	0
1	0	1	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0

Après l'application de la simplification algébrique : $S3 = 0$

4) L'expression de $S4 = \overline{(x + y). (z + t). (\bar{x} + \bar{y})}$

TV (S4)

	y	Z	T	x+y	$\overline{x+y}$	z+t	$(x+y) \cdot (\overline{x+y}) \cdot (z+t)$	S4
0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	1	0	1
0	0	1	0	0	1	1	0	1
0	0	1	1	0	1	1	0	1
0	1	0	0	1	0	0	0	1
0	1	0	1	1	0	1	0	1
0	1	1	0	1	0	1	0	1
0	1	1	1	1	0	1	0	1
1	0	0	0	1	0	0	0	1
1	0	0	1	1	0	1	0	1
1	0	1	0	1	0	1	0	1
1	0	1	1	1	0	1	0	1
1	1	0	0	1	0	0	0	1
1	1	0	1	1	0	1	0	1
1	1	1	0	1	0	1	0	1
1	1	1	1	1	0	1	0	1

Après l'application de la simplification algébrique :S4=1

5) L'expression de S5= $(x \oplus \bar{y}) + x$

X	y	\bar{y}	$(x \oplus \bar{y})$	S5
0	0	1	1	1
0	1	0	0	0
1	0	1	0	1
1	1	0	1	1

Après l'application de la simplification algébrique : $S5 = x + \bar{y}$

Exercice 3 :

1) $F1 = x + y + \bar{x} \cdot (y+z)$

x	Y	z	\bar{x}	(y + z)	$\bar{x} \cdot (y+z)$	x + y	F1
0	0	0	1	0	0	0	0
0	0	1	1	1	1	0	1
0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1
1	0	1	0	1	0	1	1
1	1	0	0	1	0	1	1
1	1	1	0	1	0	1	1

$$F1(\text{SOP}) = (\bar{x} \cdot \bar{y} \cdot z) + (\bar{x} \cdot y \cdot \bar{z}) + (\bar{x} \cdot y \cdot z) + (x \cdot \bar{y} \cdot \bar{z}) + (x \cdot \bar{y} \cdot z) + (x \cdot y \cdot \bar{z}) + (x \cdot y \cdot z)$$

$$F1(\text{POS}) = x + y + z$$

2) $F2 = ((x + y) \cdot (x + \bar{y}) + y) \cdot (\bar{x} + y)$

x	y	\bar{x}	\bar{y}	x + y	x + \bar{y}	$\bar{x} + y$	(x + y) \cdot (x + \bar{y})	(x + y) \cdot (x + \bar{y}) + y	F2
0	0	1	1	0	1	1	0	0	0
0	1	1	0	1	0	1	0	1	1
1	0	0	1	1	1	0	1	1	0
1	1	0	0	1	1	1	1	1	1

$$F2(\text{SOP}) = (\bar{x} \cdot y) + (x \cdot y)$$

$$F2(\text{POS}) = (x + y) \cdot (\bar{x} + y)$$

3) $F3 = (x + y) \cdot (\bar{y} + z) \cdot (\bar{x} + z)$

X	Y	z	\bar{x}	\bar{y}	$x + y$	$\bar{y} + z$	$\bar{x} + z$	F3
0	0	0	1	1	0	1	1	0
0	0	1	1	1	0	1	1	0
0	1	0	1	0	1	0	1	0
0	1	1	1	0	1	1	1	1
1	0	0	0	1	1	1	0	0
1	0	1	0	1	1	1	1	1
1	1	0	0	0	1	0	0	0
1	1	1	0	0	1	1	1	1

$$F3(\text{SOP}) = (\bar{x}.y.z) + (x.\bar{y}.z) + (x.y.z)$$

$$F3(\text{POS}) = (x + y + z) . (x + y + \bar{z}) . (x + \bar{y} + z) . (\bar{x} + y + z) . (\bar{x} + \bar{y} + z)$$

$$4) F4 = \overline{x + y} + (x \oplus \bar{y})$$

X	y	\bar{y}	$x + y$	$\overline{x + y}$	$(x \oplus \bar{y})$	F4
0	0	1	0	1	1	1
0	1	0	1	0	0	0
1	0	1	1	0	0	0
1	1	0	1	0	1	1

$$F4(\text{SOP}) = (\bar{x}.\bar{y}) + (x.y)$$

$$F4(\text{POS}) = (x + \bar{y}).(\bar{x} + y)$$

$$5) F5 = (x + y) . (\bar{y} + z) + \bar{t} (x + y)$$

X	y	Z	T	\bar{y}	\bar{t}	x+y	$\bar{y} + z$	$(x + y) \cdot (\bar{y} + z)$	$\bar{t} (x + y)$	F5
0	0	0	0	1	1	0	1	0	0	0
0	0	0	1	1	0	0	1	0	0	0
0	0	1	0	1	1	0	1	0	0	0
0	0	1	1	1	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	0	0	1	0	0	0	0
0	1	1	0	0	1	1	1	1	1	1
0	1	1	1	0	0	1	1	1	0	1
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	0	1	1	1	0	1
1	0	1	0	1	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1	0	1
1	1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	0	1	0	0	0	0
1	1	1	0	0	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	0	1

$$F5(SOP) = (\bar{x} \cdot y \cdot \bar{z} \cdot \bar{t}) + (\bar{x} \cdot y \cdot z \cdot \bar{t}) + (\bar{x} \cdot y \cdot z \cdot t) + (x \cdot \bar{y} \cdot \bar{z} \cdot \bar{t}) + (x \cdot \bar{y} \cdot \bar{z} \cdot t) + (x \cdot \bar{y} \cdot z \cdot \bar{t}) + (x \cdot \bar{y} \cdot z \cdot t) + (x \cdot y \cdot \bar{z} \cdot \bar{t}) + (x \cdot y \cdot z \cdot \bar{t}) + (x \cdot y \cdot z \cdot t)$$

$$F5(POS) = (x + y + z + t) \cdot (x + y + z + \bar{t}) \cdot (x + y + \bar{z} + t) \cdot (x + y + \bar{z} + \bar{t}) \cdot (x + \bar{y} + z + \bar{t}) \cdot (\bar{x} + \bar{y} + z + \bar{t})$$

Exercice 4 :

1) Table de karnaugh à 2 variables A et B

A \ B	0	1
0	1	1
1	0	0

$F1 = \bar{B}$

A \ B	0	1
0	1	0
1	1	1

$F2 = \bar{A} + B$

A \ B	0	1
0	0	1
1	1	0

$F3 = (A+B) \cdot (\bar{A} + \bar{B})$

A \ B	0	1
0	0	0
1	1	0

$F4 = \bar{A} \cdot B$

2) Table de karnaugh à 3 variables A , B et C

AB \ C	00	01	11	10
0	1	1	0	1
1	0	0	1	1

$F5 = \bar{A} \cdot \bar{C} + A \cdot \bar{B} + A \cdot C$

AB \ C	00	01	11	10
0	0	1	1	0
1	1	1	1	1

$F6 = B + C$

AB \ C	00	01	11	10
0	1	1	0	0
1	1	0	1	0

$$F7 = (\bar{A} + C) \cdot (\bar{A} + B) \cdot (A + \bar{B} + \bar{C})$$

AB \ C	00	01	11	10
0	0	1	1	0
1	0	0	1	0

$$F8 = B \cdot (A + \bar{C})$$

3) Table de karnaugh à 4 variables A , B C et D

AB \ CD	00	01	11	10
00	0	1	1	1
01	0	0	0	1
11	1	0	1	0
10	0	1	1	0

AB \ CD	00	01	11	10
00	1	1	1	0
01	1	0	1	1
11	1	1	1	0
10	1	0	1	1

$$F9 = B \cdot \bar{D} + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot C + \bar{A} \cdot \bar{B} \cdot C \cdot D \quad F10 = \bar{A} \cdot \bar{B} + A \cdot B + B \cdot \bar{C} \cdot \bar{D} + B \cdot C \cdot D + A \cdot \bar{C} \cdot D + A \cdot C \cdot \bar{D}$$

AB \ CD	00	01	11	10
00	0	1	0	0
01	1	1	1	0
11	1	1	1	0
10	0	0	1	0

AB \ CD	00	01	11	10
00	1	0	1	1
01	1	0	0	0
11	1	0	0	0
10	1	1	0	0

$$F11 = (B + D) \cdot (\bar{A} + B) \cdot (\bar{A} + C + D) \cdot (A + \bar{C} + D)$$

$$F12 = (A + \bar{B} + C) \cdot (\bar{B} + \bar{D}) \cdot (\bar{A} + \bar{D}) \cdot (\bar{A} + \bar{C})$$

Exercice 5 :

Simplifier les tables de karnaugh suivantes avec des valeurs incomplètes.

AB \ C	00	01	11	10
0	0	x	1	0
1	X	1	1	1

$$F1 = B + C$$

AB \ C	00	01	11	10
0	x	x	0	1
1	x	1	1	1

$$F2 = \bar{B} + C$$

AB \ CD	00	01	11	10
00	0	x	0	x
01	0	0	1	0
11	X	x	x	1
10	0	0	1	0

$$F3 = C.D + ABD + ABC$$

AB \ CD	00	01	11	10
00	0	0	1	1
01	x	x	0	1
11	x	1	0	0
10	0	0	x	1

$$F2 = A.\bar{D} + \bar{B}.\bar{C}.D + \bar{A}.D$$

2 Chapitre 2

Exercice 1:

On donne ci-dessous la table de transcodage suivante :

Entrées			Sorties		
A3	A2	A1	B3	B2	B1
0	0	0	0	1	1
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	0	1
1	1	0	0	1	0
1	1	1	1	0	0

1. Donner la deuxième forme canonique (POS) de la fonction B1 .

$$B1_{(POS)} = (A3 + \overline{A2} + A1) \cdot (\overline{A3} + \overline{A2} + A1) \cdot (\overline{A3} + \overline{A2} + \overline{A1})$$

2. Donner la première forme canonique (SOP) des fonctions B2 et B3 .

$$B2_{(SOP)} = \overline{A3} \cdot \overline{A2} \cdot \overline{A1} + \overline{A3} \cdot A2 \cdot \overline{A1} + \overline{A3} \cdot A2 \cdot A1 + A3 \cdot A2 \cdot \overline{A1}$$

$$B3_{(SOP)} = \overline{A3} \cdot A2 \cdot \overline{A1} + \overline{A3} \cdot A2 \cdot A1 + A3 \cdot A2 \cdot A1$$

3. Simplifier les sorties du transcodeur (B3, B2, B1) par la méthode de karnaugh.

A3A2 \ A1	00	01	11	10
0	1	0	0	1
1	1	1	0	1

$$B1 = \overline{A2} + \overline{A3} \cdot A1$$

A3A2 \ A1	00	01	11	10
0	1	1	1	0
1	0	1	0	0

$$B2 = \overline{A3} \cdot \overline{A1} + A2 \cdot \overline{A1} + \overline{A3} \cdot A2$$

A3A2 A1	00	01	11	10
0	0	1	0	0
1	0	1	1	0

$$B3 = \overline{A3} \cdot A2 + A2 \cdot A1$$

4. Donner les circuits logiques des fonctions simplifiées à l'aide des portes NAND.

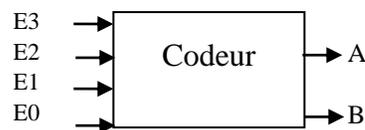
$$B1 = \overline{A2 \cdot \overline{A3} \cdot \overline{A3} \cdot A1}$$

$$B2 = \overline{\overline{A3} \cdot \overline{A3} \cdot \overline{A1} \cdot \overline{A1} \cdot A2 \cdot \overline{A1} \cdot \overline{A1} \cdot \overline{A3} \cdot \overline{A3} \cdot A2}$$

$$B3 = \overline{\overline{A3} \cdot \overline{A3} \cdot A2 \cdot \overline{A2} \cdot A1}$$

Exercice 2:

Soit un circuit combinatoire à 4 lignes d'entrée et 2 lignes de sorties comme le montre la figure ci-dessous : Le fonctionnement est le suivant :



- Lorsqu'une seule ligne d'entrée, parmi E0, E1, E2, E3 se trouve au niveau haut (=1), son numéro est codé en binaire sur les sorties (BA),
- Si plusieurs lignes sont simultanément au niveau haut (=1), on code le numéro le plus élevé,
- Si toutes les lignes d'entrée sont au niveau bas (=0), on code (BA) = (00).

1) Dresser la table de vérité du codeur.

E0	E1	E2	E3	B	A
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	1	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	1	0
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	0
1	1	1	1	1	1

Etablir les équations logiques des sorties A, B en fonction des entrées de E0...E3

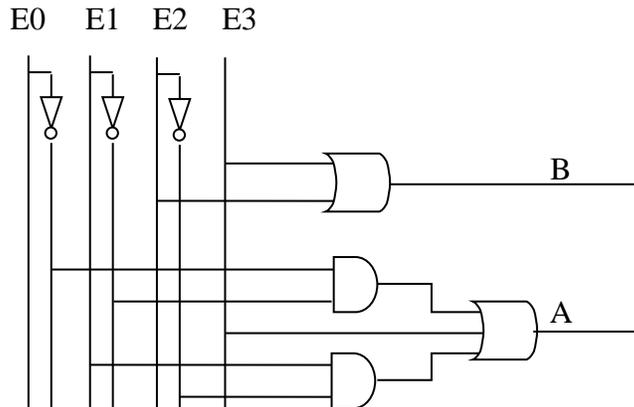
E0E1	00	01	11	10
E2E3				
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

E0E1	00	01	11	10
E2E3				
00	1	1	1	0
01	1	1	1	1
11	1	1	1	1
10	1	0	0	0

$$B = E3 + E2$$

$$A = \overline{E0} \cdot \overline{E1} + E3 + E1 \cdot \overline{E2}$$

1) Représenter le schéma logique du codeur.



Exercice 3 :

Trois interrupteur I1, I2 et I3 commandent le démarrage de deux moteurs M1 et M2 selon les conditions suivantes :

Le moteur M1 ne doit démarrer que si au moins deux interrupteurs sont activés (=1),

Dés qu'un ou plusieurs interrupteurs sont activés, le moteur M2 doit démarrer.

1) Donner la table de vérité correspondante,

I1	I2	I3	M1	M2
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

2) Déterminer les équations simplifiées des sorties M1 et M2 par la méthode de karnaugh.

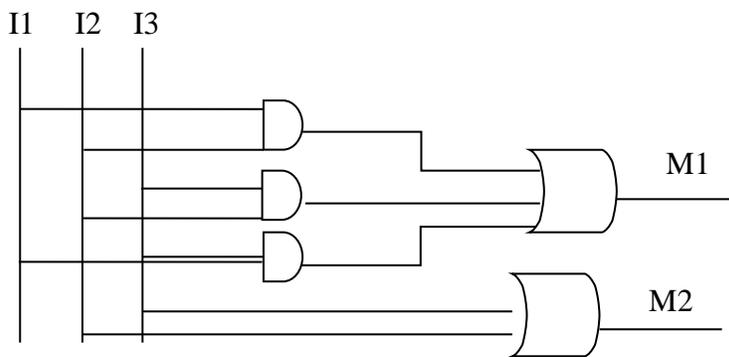
I1I2 \ I3	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$M1 = I1 \cdot I2 + I2 \cdot I3 + I1 \cdot I3$$

I1I2 \ I3	00	01	11	10
0	0	1	1	1
1	1	1	1	1

$$M2 = I1 + I2 + I3$$

3) Faire le logigramme des sorties M1 et M2 simplifiées.



Exercice 4

Soient a,b,c, d des variables logiques, on considère les fonctions logiques F et G définies par :

$$F = 1 \text{ si et seulement si } a+b \leq c+d$$

$$G = 1 \text{ si et seulement si la combinaison } abcd \text{ est strictement inférieure à } 2(10).$$

1. Dresser la table de vérité de ces fonctions.

A	B	C	d	a+b	c+d	F	G
0	0	0	0	0	0	1	1
0	0	0	1	0	1	1	1
0	0	1	0	0	1	1	0
0	0	1	1	0	1	1	0
0	1	0	0	1	0	0	0
0	1	0	1	1	1	1	0
0	1	1	0	1	1	1	0
0	1	1	1	1	1	1	0
1	0	0	0	1	0	0	0
1	0	0	1	1	1	1	0
1	0	1	0	1	1	1	0
1	0	1	1	1	1	1	0
1	1	0	0	1	0	0	0
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0

2. Donner la deuxième forme canonique de F(POS) et la première forme canonique de G (SOP).

$$F(\text{POS}) = (a + \bar{b} + c + d) \cdot (\bar{a} + b + c + d) \cdot (\bar{a} + \bar{b} + c + d)$$

$$G(\text{SOP}) = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d$$

3. Simplifier les fonctions de sorties F et G par la méthode de karnaugh.

a b \ cd	00	01	11	10
00	1	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$F = c + d + \bar{a} \cdot \bar{b}$$

ab \ cd	00	01	11	10
00	1	0	0	0
01	1	0	0	0
11	0	0	0	0
10	0	0	0	0

$$A = \bar{a} \cdot \bar{b} \cdot \bar{c}$$

Exercice 5

Soit un circuit combinatoire à 04 variables d'entrées A,B,C,D et 03 sorties S1, S2 et S3 telles que :

S1 est vraie si (A est vraie) et (deux autres variables au moins sont vraies),

S2 est vraie si (seule A est fausse) ou (A est vraie et deux autres variables au moins sont fausses).

S3 est vraie si (A est fausse et une autre variable au moins est fausse).

1) Etablir la table de vérité de ce circuit (Entrées + sorties).

A	B	C	D	S1	S2	S3
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	0	1
0	1	0	1	0	0	1
0	1	1	0	0	0	1
0	1	1	1	0	1	0
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	1	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	0	0

2) Ecrire les équations simplifiées (par tableau de Karnaugh) des sorties S1, S2 et S3.

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	1	1
10	0	0	1	0

$$S1 = A.B.D + A.B.C + A.C.D$$

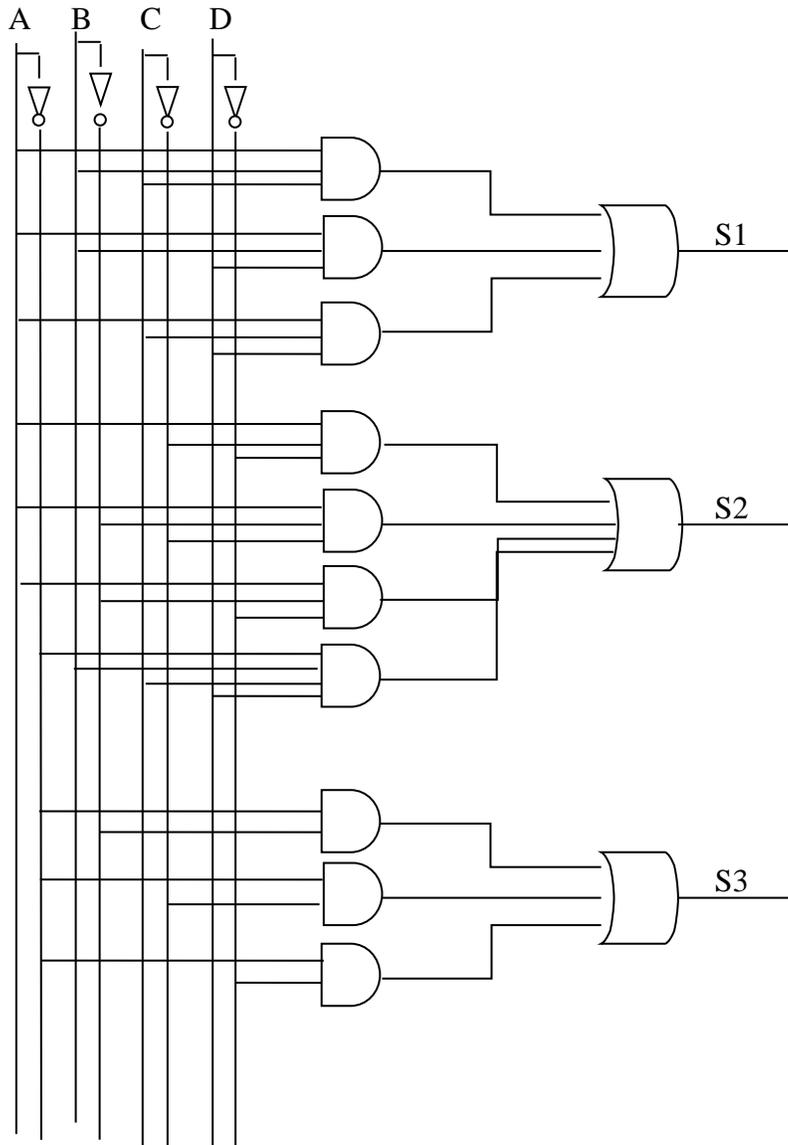
AB \ CD	00	01	11	10
00	0	0	1	1
01	0	0	0	1
11	0	1	0	0
10	0	0	0	1

$$S2 = A..C..D + A..B..C + A..B..D + \bar{A}.B.C.D$$

AB \ CD	00	01	11	10
00	1	1	0	0
01	1	1	0	0
11	1	0	0	0
10	1	1	0	0

$$S3 = \bar{A}.\bar{C} + \bar{A}.\bar{B} + \bar{A}.\bar{D}$$

3) Dessiner le logigramme des sorties simplifiées S1, S2 et S3.

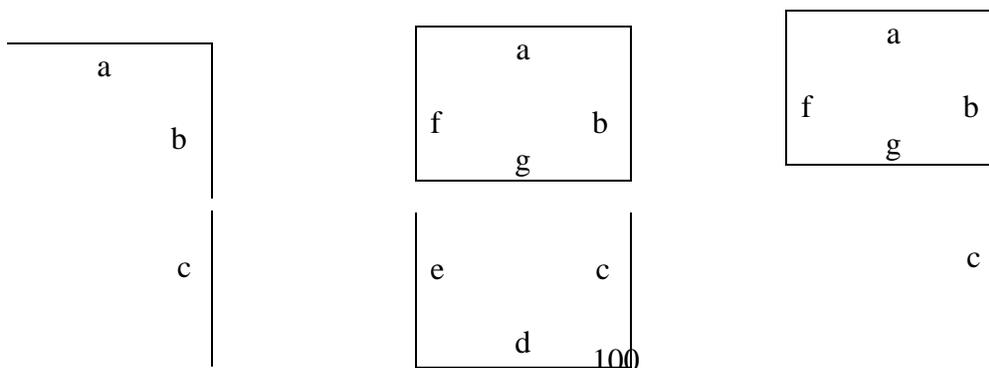
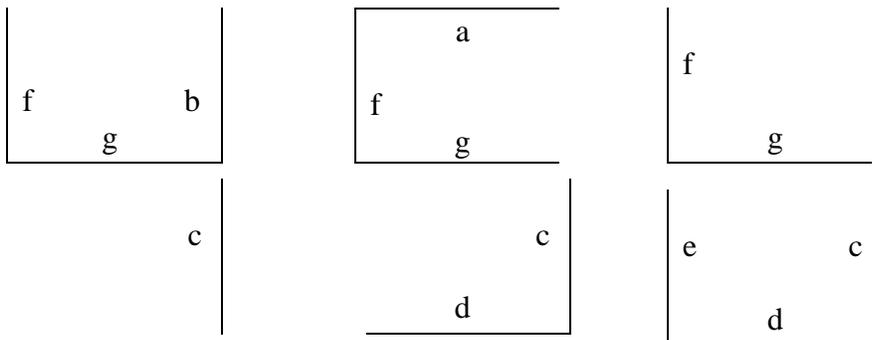
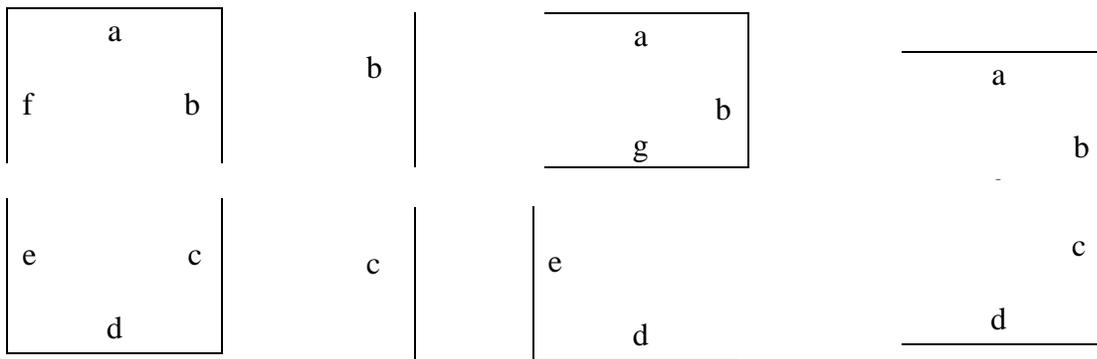


Exercice 6 : Afficheur 7 segments

Donner la table de vérité d'un décodeur BCD - 7 segments qui permet d'afficher les 10 chiffres décimaux à l'aide de 7 segments, notés de a à g. un segment prend la valeur 0 s'il est éteint ou 1 s'il est allumé.

Solution

Il s'agit de décoder un nombre en code BCD sur 4 bits en son équivalent décimale. Le chiffre décimal est affiché à l'aide des segments. Ci-dessous les segments allumés correspondants aux chiffres décimaux.



La table de vérité du décodeur BCD – 7 segments contient 4 entrées $e_3e_2e_1e_0$ qui représentent le code BCD et 7 sorties qui représentent les segments a, b, c, d, e, f et g. un segment ayant la valeur 1 indique un segment allumé et 0 indique un segment éteint. La table de vérité de ce décodeur est la suivante :

e_3	e_2	e_1	e_0		a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	0	1	1	0	0	0	0
0	0	1	0	2	1	1	0	1	1	0	1
0	0	1	1	3	1	1	1	1	0	0	1
0	1	0	0	4	0	1	1	0	0	1	1
0	1	0	1	5	1	0	1	1	0	1	1
0	1	1	0	6	0	0	1	1	1	1	1
0	1	1	1	7	1	1	1	0	0	0	0
1	0	0	0	8	1	1	1	1	1	1	1
1	0	0	1	9	1	1	1	0	0	1	1

Les équations des sorties a, b, c, d, e, f et g en fonction des entrées.

Les équations du décodeur sont obtenues à l'aide des tables de karnaugh. On aura donc 7 tableaux de karnaugh à simplifier.

e_3e_2 e_1e_0	00	01	11	10
00	1	0	x	1
01	0	1	x	1
11	1	1	x	x
10	1	0	x	x

$$a = (\overline{e_2} + e_0) \cdot (e_3 + e_2 + e_1 + \overline{e_0})$$

e_3e_2 e_1e_0	00	01	11	10
00	1	1	x	1
01	1	0	x	1
11	1	1	x	x
10	1	0	x	x

$$b = (\overline{e_2} + e_1 + \overline{e_0}) \cdot (\overline{e_2} + \overline{e_1} + e_0)$$

e3e2 \ e1e0	00	01	11	10
00	1	1	x	1
01	1	1	x	1
11	1	1	x	x
10	0	1	x	x

e3e2 \ e1e0	00	01	11	10
00	1	0	X	1
01	0	1	X	0
11	1	0	X	x
10	1	1	X	x

$$c = e2 + \overline{e1} + e0$$

$$d = (\overline{e2} + e1 + e0) \cdot (\overline{e2} + \overline{e1} + \overline{e0}) \cdot (e2 + e1 + \overline{e0})$$

e3e2 \ e1e0	00	01	11	10
00	1	0	x	1
01	0	0	x	0
11	0	0	x	x
10	1	1	x	x

e3e2 \ e1e0	00	01	11	10
00	1	1	x	1
01	0	1	x	1
11	0	0	x	x
10	0	1	x	x

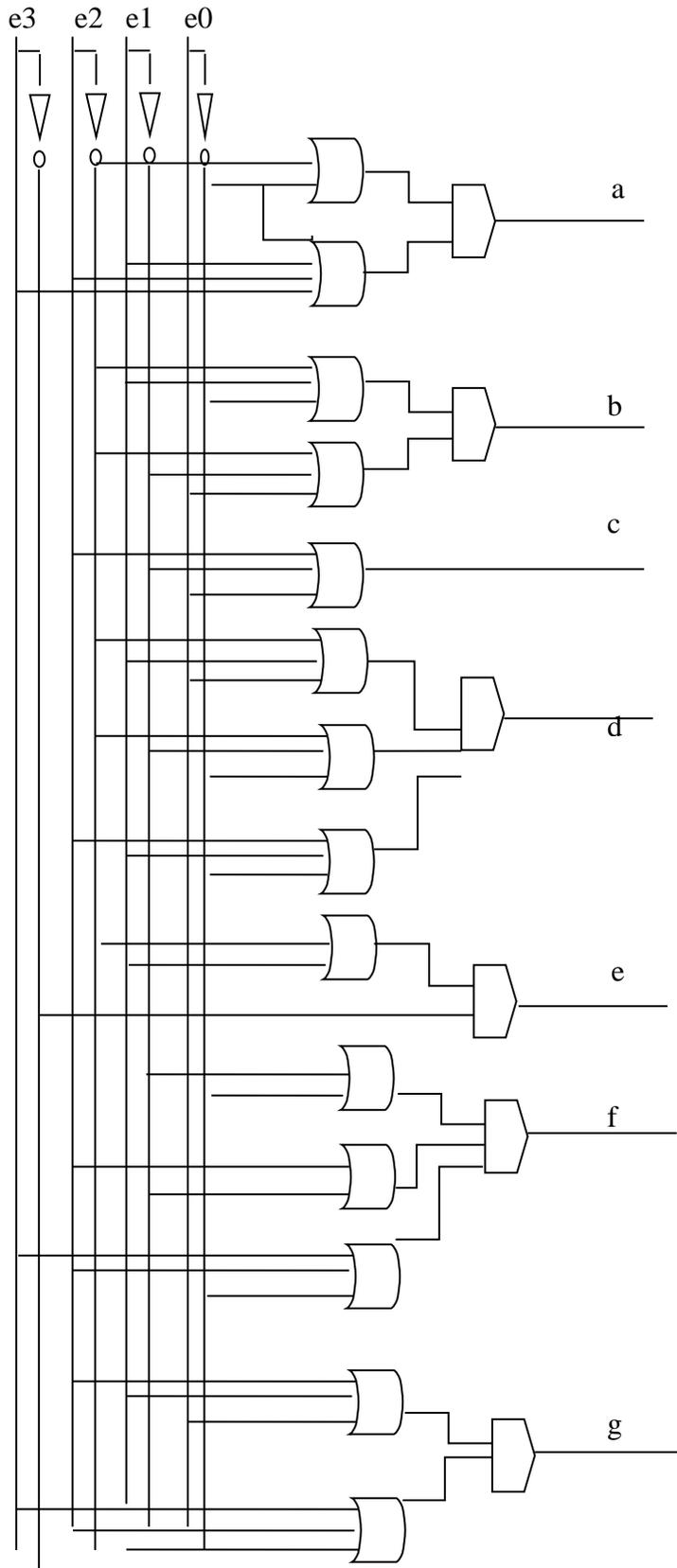
$$e = \overline{e0} \cdot (\overline{e2} + e1)$$

$$f = (\overline{e1} + \overline{e0}) \cdot (e2 + \overline{e1}) \cdot (e3 + e2 + \overline{e0})$$

e3e2 \ e1e0	00	01	11	10
00	0	1	x	1
01	0	1	x	1
11	1	0	x	x
10	1	1	x	x

$$g = (\overline{e2} + \overline{e1} + \overline{e0}) \cdot (e3 + e2 + e1)$$

Le logigramme

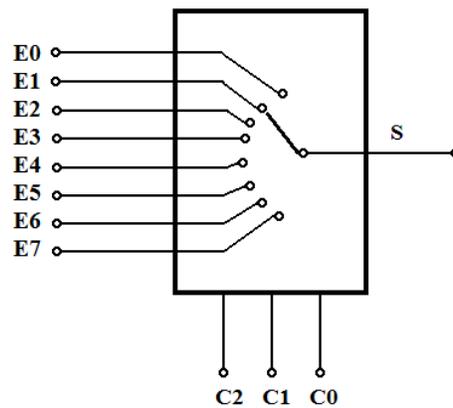


Exercice 7 :

Donner la fonction algébrique d'un multiplexeur 8 vers 1.

Corrigé

Un multiplexeur 8 vers 1 « 8 vers 1 » contenant 8 entrées (E0, E1,..E7) et une sortie S, nécessite trois (3) bits de sélection ou de commande (C2, C1, C0) car $8 = 2^3$. Son schéma est le suivant :



Sa fonction algébrique sera déduite à partir de sa table de vérité qui est donnée par :

C2	C1	C0	S
0	0	0	E0
0	0	1	E1
0	1	0	E2
0	1	1	E3
1	0	0	E4
1	0	1	E5
1	1	0	E6
1	1	1	E7

La forme Sop de la fonction S est extraite à partir de la table de vérité :

$$S = \overline{C2}.\overline{C1}.\overline{C0}.(E0) + \overline{C2}.\overline{C1}.C0(E1) + \overline{C2}.C1.\overline{C0}(E2) + \overline{C2}.C1.C0(E3) + C2.\overline{C1}.\overline{C0}(E4) + C2.\overline{C1}.C0(E5) + C2.C1.\overline{C0}(E6) + C2.C1.C0(E7)$$

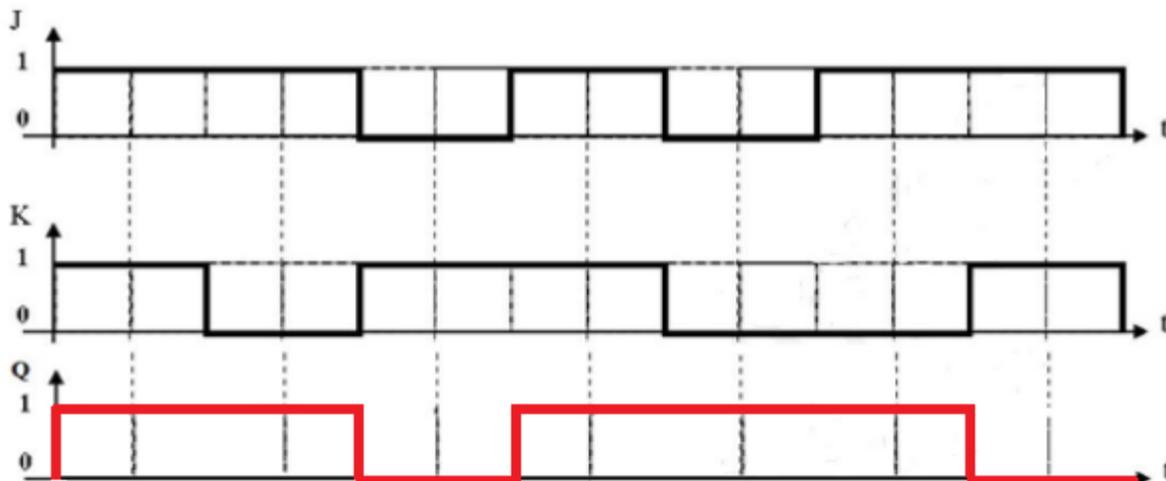
3 Chapitre 3

Exercice 1:

1/ Donner la TV de la bascule JK asynchrone.

J	K	Q_n	\bar{Q}_n
0	0	Q_{n-1}	\bar{Q}_{n-1}
0	1	0	1
1	0	1	0
1	1	\bar{Q}_{n-1}	Q_{n-1}

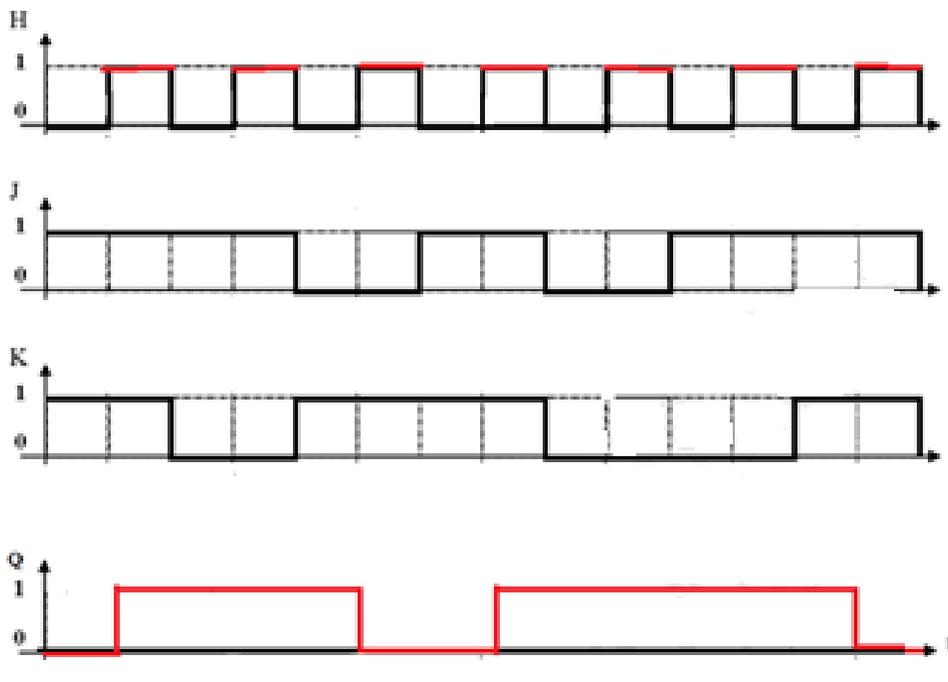
2/ Compléter le chronogramme de la bascule JK asynchrone en fonction des entrées sachant que initialement la sortie Q est mise à zéro ($Q=0$).



3/ Donner la TV de la bascule JKH synchrone sur le niveau haut.

Les entrées				Sortie	Etat
H	J	K	Q_{n-1}	Q_n	
0	∇	∇	Q_{n-1}	Q_{n-1}	Mémorisation
0 (Niveau bas)	∇	∇	Q_{n-1}	Q_{n-1}	Mémorisation
1 (Niveau haut)	0	0	Q_{n-1}	Q_{n-1}	Mémorisation
1 (Niveau haut)	0	1	Q_{n-1}	0	Mise à zero
1 (Niveau haut)	1	0	Q_{n-1}	1	Mise à un
1 (Niveau haut)	1	1	Q_{n-1}	Q_{n-1}	\bar{Q}_{n-1}

4/ Compléter le chronogramme de la bascule JKH synchrone sur le niveau haut.



5/ La durée entre deux niveaux successifs s'appelle période (T), donner la fréquence de cette horloge en Hz puis en Mhz sachant que la période =2ms.

$$\text{Période (T)} = 2 \text{ ms} = 2 \cdot 10^{-3} \text{ sec}$$

$$\text{Fréquence} = 1/T = 1/2 \cdot 10^{-3} = 0.5 \cdot 10^3 \text{ Hz} = 0.5 \cdot 10^{-3} \text{ Mhz} \text{ puisque } 1 \text{ Mhz} = 10^6 \text{ Hz.}$$

Exercice 3 :

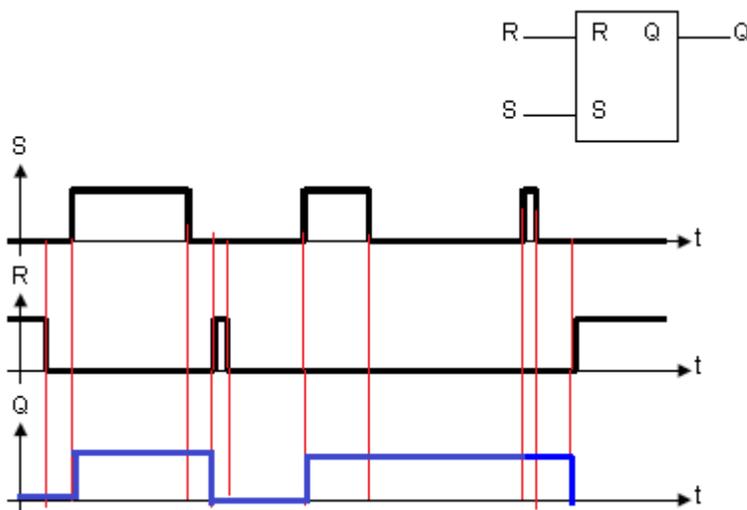
Compléter le chronogramme ci-dessous.

Il s'agit de définir les valeurs de l'état de sortie Q de la bascule RS asynchrone puisque l'horloge H n'est pas considérée comme une entrée de ce circuit séquentiel.

Rappelons la table de vérité de la bascule RS asynchrone .

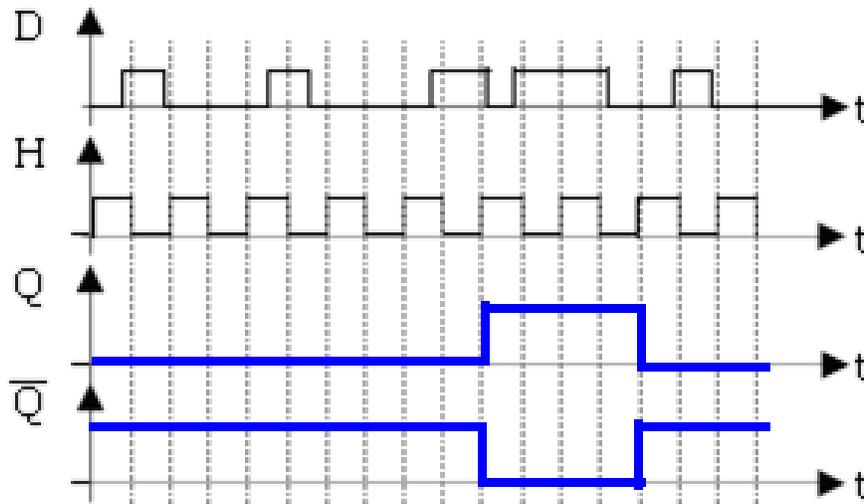
R	S	Q_n	\bar{Q}_n
0	0	Q_{n-1}	\bar{Q}_{n-1}
0	1	1	0
1	0	0	1
1	1	ϕ	ϕ

On commence d'abord par marquer les variations de chaque entrée R et S dans le chronogramme, puis pour chaque changement des entrées, on trace le chronogramme de l'état de sortie Q en se basant sur la table de vérité de RS donnée ci-dessus. Le chronogramme de l'état de Q est donc le suivant



Exercice 4:

Compléter les chronogrammes suivants (appliqués à une bascule D active sur front montant) :



Rappelons le fonctionnement de la bascule D. La table de vérité de la bascule D synchrone sur niveau haut est donnée comme suit :

Entrées		Sortie
H	D	Q_n
0	\forall	Q_{n-1}
1	0	0
1	1	1

Rappelons que l'état Q reste stable entre les deux fronts successifs.

Nous remarquons que dans le premier front montant $D=0$. La sortie Q reste à l'état 0 jusqu'au prochain front montant.

Dans le deuxième front montant, $D = 0$ ce qui implique que la sortie Q reste toujours à l'état 0 et elle va garder cette valeur jusqu'au prochain front montant et ainsi de suite.

Dans le sixième front montant, $D = 1$, ce qui déclenche un basculement de la sortie Q ($Q = \bar{0} = 1$). La sortie reste à l'état 1 jusqu'au 8^{ème} front montant où l'état de sortie Q devient 1 car à cet instant $D=1$.

Exercice 5:

Donner les Tables de transition des bascule RS, JK et D.

Rappel :

La table de transition d'une bascule définie les valeurs des entrées synchrones qui ont provoqués le basculement. Il s'agit de déterminer les valeurs des entrées (RS, JK et D) sachant qu'on connaît déjà les valeurs des sorties

Réponse : la table de transition des bascule RS, JK et D est donnée par le tableau suivant :

Q_{n-1}	Q_n	J	K	R	S	D
0	0	0	X	X	0	0
0	1	1	X	0	1	1
1	0	X	1	1	0	0
1	1	X	0	0	X	1

Bacule RS

La question est de trouver les valeurs RS pour les différentes valeurs de Q_{n-1} et Q_n

Pour la première ligne, nous avons $Q_{n-1} = Q_n = 0$

Nous avons dans ce cas deux états possibles de la bascule JK:

1. Soit une mémorisation car $Q_{n-1} = Q_n$ et dans ce cas $R=0$ et $S=0$
2. Soit une mise à zero puisque $Q_n=0$ et dans ce cas $R=1$ et $S=0$

Nous remarquons que pour les deux cas S est toujours égale à 0 (donc nous remplissons dans le champs correspondant la valeur 0). la valeur X est assigné au champs R pour dire que ce dernier peut prendre deux valeurs 0 et 1 ($x=\{0,1\}$).

Nous remplissons les autres champs en utilisant le même raisonnement

Bacule JK

La question est de trouver les valeurs JK pour les différentes valeurs de Q_{n-1} et Q_n

Pour la première ligne, nous avons $Q_{n-1} = Q_n = 0$

Nous avons dans ce cas deux états possibles de la bascule JK:

1. Soit une mémorisation car $Q_{n-1} = Q_n$ et dans ce cas $J=0$ et $K=0$
2. Soit une mise à zero puisque $Q_n=0$ et dans ce cas $J=0$ et $K=1$

Nous remarquons que pour les deux cas J est toujours égale à 0 (donc nous remplissons dans le champs correspondant la valeur 0). la valeur X est assigné au champs K pour dire que ce dernier peut prendre deux valeurs 0 et 1 ($x=\{0,1\}$).

Nous remplissons les autres champs en utilisant le même raisonnement.

Bacule D

Dans la bascule D, la sortie Q_n suit l'entrée D, par conséquent $Q_n=D$.

Exercice 6 :

Faites la réalisation d'un compteur synchrone modulo 5 (cycle incomplet), en utilisant des bascules D à front montant.

Le compteur modulo 5 suit un cycle incomplet car il permet de compter de (000) jusqu'à (100) puis revient à (000) et donc les combinaisons 101 et 111 ne sont pas prises en compte par ce compteur.

Ce compteur synchrone nécessite donc 3 bascules pour le réaliser : D1, D2 et D3. Dans ce cas toutes les bascules seront reliées au même signal d'horloge.

Soit Q_i =état précédent de la bascule D_i et Q_i^+ = état actuel de la bascule D_i . La table de transition des trois bascules D_1 , D_2 et D_3 prises en même temps est la suivante :

Q_3	Q_2	Q_1	Q_2^+	Q_2^+	Q_1^+	D3	D2	D1
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	0	0	0	0	0	0

Les fonctions logiques de D1, D2 et D3 en fonction des états Q3, Q2 et Q1 sont calculées par les 3 tableaux de karnaugh suivants :

Q3Q2 \ Q1	00	01	11	10
0	1	1	x	0
1	0	0	x	x

$$D1 = \overline{Q1} \cdot \overline{Q3}$$

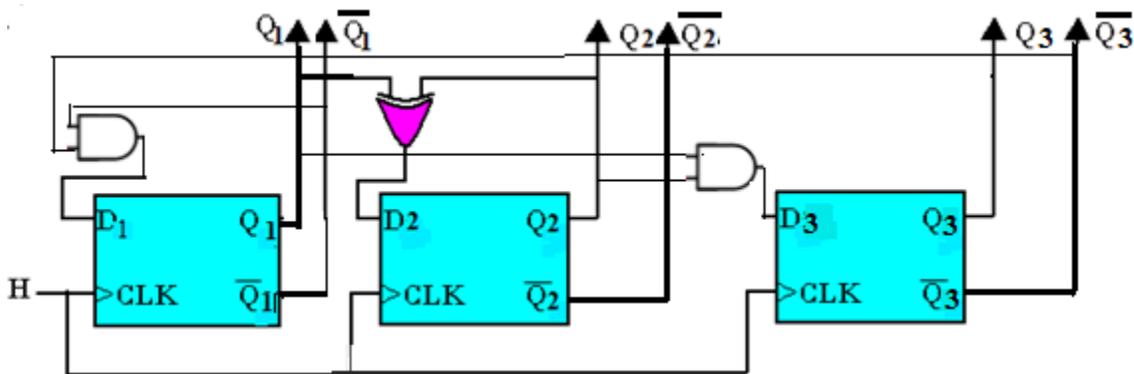
Q3Q2 \ Q1	00	01	11	10
0	0	1	x	0
1	1	0	x	x

$$D2 = Q2 \cdot \overline{Q1} + Q1 \cdot \overline{Q2} = Q1 \oplus Q2$$

Q3Q2 \ Q1	00	01	11	10
0	0	0	x	0
1	0	1	x	x

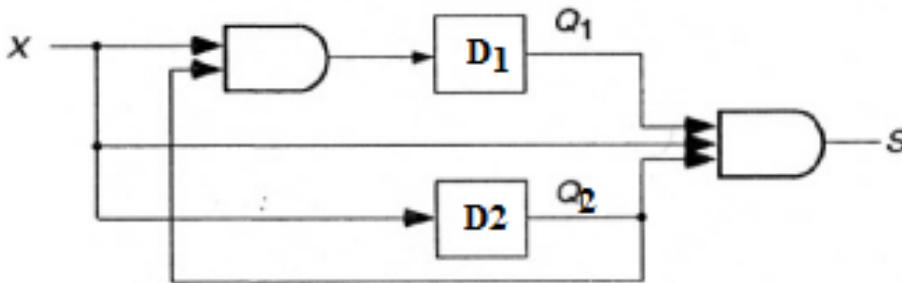
$$D3 = Q1 \cdot Q2$$

Le circuit séquentiel du compteur synchrone modulo 5 est donné par le schéma suivant :



Exercice7 :

Analyser le circuit suivant :



1) Déterminer la table des états

Etape1 :

Exprimons d'abord les entrées des bascules (D1 et D2) en fonction de l'entrée X et les états précédents (Q1 et Q2) :

$$D1 = X \cdot Q2$$

$$D2 = X$$

Etape2 :

Exprimons les sorties des bascules en fonction de l'entrée X et les états précédents (Q1 et Q2). Notons pour la bascule D, l'état futur (Q^+) est égal à l'entrée ($Q^+ = D$).

Donc pour chaque bascule D, nous avons : $Q_1^+ = D_1$ et $Q_2^+ = D_2$

Ce qui donne :

$$Q_1^+ = D_1 = X \cdot Q_2$$

$$Q_2^+ = D_2 = X$$

Etape3 :

Déterminons maintenant la sortie du circuit (Y) :

$$Y=X.Q1.Q2$$

Nous remarquons que les valeurs des variables de sortie dépendent des états présents Q1 et Q2 et la variable d'entrée X. le système séquentiel est donc appelé une machine de Mealy. On a donc $Y = f(X, Q1, 2)$

Etape4 :

Traçons maintenant la table de vérité (table d'état) de ce circuit à trois entrées (X,Q1,Q2):

Entrée X	Etats précédents		Etats futurs		Sortie Y
	Q1	Q2	Q1⁺	Q2⁺	
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	1	0
1	0	1	1	1	1
1	1	0	0	1	0
1	1	1	1	1	1

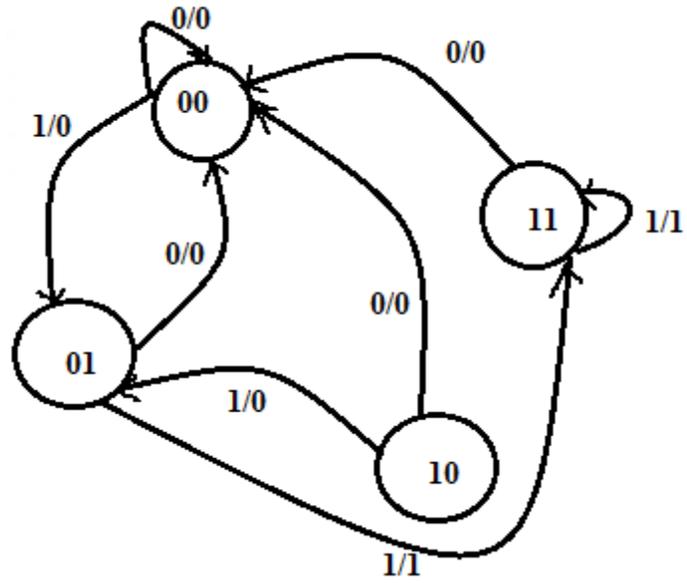
Etape5 : diagramme de transition.

Traçons maintenant le diagramme de transition.

Dans ce circuit nous avons deux bascules donc il y a $2^2=4$ états possibles :

Etat0 =00, Etat 1=01, Etat2=10 et Etat3=11

On obtient donc le graphe de transition suivant :



Références :

- [1] Robert. Strandh et al., Architectures des l'ordinateurs. Edition : DUNOD 2005.
- [2] Dounia Yedjour, Codification et Représentation de l'Information, Polycopié de cours, Faculté de Maths-info, Département d'informatique, université de USTO, Oran, 2017.
- [3] J.M. Muller, Arithmétique des ordinateurs, opérateurs et fonctions élémentaires, Etudes et recherches en informatique, Masson, 1989.
- [4] Phillipe Darch. Logique booléenne et implémentation Technologique. Edition VUIBERT : 2004.
- [5] John R. Gregg, Ones and Zeros: Understanding Boolean Algebra, Digital Circuits, and the Logic of Sets 1st Edition , Wiley & sons Inc. publishing, 1998, ISBN: 978-0-7803-3426-7.
- [6] Bradford Henry Arnold , Logic and Boolean Algebra, Dover publication, Inc., Mineola, New York, 2011, ISBN-13: 978-0-486-48385-6.
- [7] Emanuel Lazard. Architectures des l'ordinateurs. Edition : PEARSON EDUCATION 2006.
- [8] Tanenbaum Andrew. Architectures des l'ordinateurs. Edition : PEARSON EDUCATION 2005.
- [9] Jean Jacques et al. Architectures des l'ordinateurs. Edition : EYROLLES 2005.
- [10] Alain Cazes, Joëlle Delacroix, architecture des machines et des systèmes informatiques : Cours et exercices corrigés, 3^o édition, Dunod 2008.
- [11] [M. Sbaï, Electronique numérique, logique combinatoire et composants numérique, Ellipses Editions Marketing, Paris, France, 2013.
- [12] A. Tachet, Automatismes, Tome 1, logique combinatoire, Office des Publications Universitaires, Alger, Algérie, 1988.
- [13] Didier Muller, Chapitre 3: Codage de l'information,
<https://www.apprendre-en-ligne.net/info/codage/codage.pdf,2018>.

- [14] JG. Dumas et al., Théorie des codes : Compression, cryptage, correction, Dunod, 2006.
- [15] Martin B., Codage, cryptologie et applications, Presses Polytechniques et Universitaires Romandes (PPUR), 2004.
- [16] Hocine Amimeur, Logique Combinatoire et Séquentielle, Polycopié de cours, Faculté de Technologie, Département de Génie Electrique, université de Bejaia, 2016.
- [17] Mc. Belaid et collectif, Logique combinatoire et séquentielle, Presses de Mitidja, Baraki, Alger, Algérie, 2010.
- [18] J. J. Mercier, Computers 2, séquence après séquence, logique séquentielle, Ellipses Editions Marketing, Paris, France, 2006.